Improving Energy Estimation at NOvA with Recurrent Neural Networks

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE
UNIVERSITY OF MINNESOTA
BY

Dmitrii Torbunov

IN PARTIAL FULFILLMENT OF THE REQUIERMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Gregory Pawloski

May, 2021

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The existence of a new elementary particle that we know now as the neutrino was proposed back in 1930 by Wolfgang Pauli [1] in order to explain apparently missing energy in the beta decays. It took more than 20 years before the neutrino was finally discovered in 1956 in a Cowan–Reines neutrino experiment [1]. At the moment of writing this thesis, it has been more than 60 years since the discovery of neutrinos. Still, we have little information known about these particles.

We know that there are three flavors of neutrinos corresponding to the three generations of leptons. The neutrinos are electrically neutral particles and do not participate in the Strong interactions. They can only interact via the Weak and Gravitational forces and therefore they are very difficult to detect. The observed neutrinos have very small but nonzero masses, less than a couple of eVs [1]. However, we currently do not have direct measurements of these masses.

The small interaction rate of neutrinos means that they can travel enormous interstellar distances without significant scattering and energy loss. This opens the possibility of using neutrinos produced in stars in order to investigate their structure in an emerging area known as neutrino astronomy [2] . Unlike photons that can be easily absorbed by plasma and interstellar gas clouds, neutrinos do not have an electric charge and travel virtually free from the interior of the stars to the observer. Additionally, small masses of neutrinos allow them to carry information about the stars at speeds close to the speed of light.

Neutrinos were observed to periodically change their flavor as they travel from

the source to the observer in a phenomenon known as the neutrino oscillations. By studying the neutrino oscillations one can get a glimpse into the very structure of the fundamental particle interactions. And, in particular, get insights about why the observable universe is predominantly made of matter and contains very little antimatter [3].

The NOvA experiment was constructed with the goal of studying the neutrino oscillation phenomenon and measuring parameters that govern neutrino oscillations with high precision. I have been a part of the NOvA experiment and was involved in the Data Analysis efforts. During my work on the NOvA experiment, I have designed a number of reconstruction algorithms based on Machine and Deep Learning techniques. In particular, I have developed a new neutrino energy estimation and a new event classification algorithms that could significantly improve the performance of the NOvA experiment. In this thesis, I will discuss my work related to the development of these algorithms.

The thesis is going to have the following structure:

- In chapter 2 I will discuss the neutrino oscillation phenomenon, its physics and describe the status of the neutrino experiments that aim at studying this phenomenon.
- In chapter 3 I will introduce the NOvA experiment and describe its design and goals.
- In chapter 4 I will begin the discussion of the data analysis at the NOvA experiment. This chapter will describe the way the NOvA detectors are calibrated and the way simulated events are generated.
- In chapter 5 I will continue the discussion of the data analysis at NOvA and discuss the way we reconstruct high-level event information from the scintillator light in the detectors.
- In chapter 6 I will conclude the discussion of the data analysis at the NOvA experiment and show how the neutrino oscillation parameters can be inferred from the measured scintillator light.

- In chapter 7 I will present my work related to the development of new Deep Learning energy estimation and event classification algorithms for the NOvA experiment.
- Finally, in chapter 8 I will show the degree of improvement that the usage of the new algorithms can provide for NOvA.

Chapter 2

Neutrino Oscillations

In order to set the stage for the NOvA experiment, I will dedicate the first chapter of my thesis to the discussion of the neutrino oscillation physics. This discussion will begin with a review of the neutrino oscillation mechanism. Then, I will summarize the current state of the neutrino oscillation experiments. Finally, I will list a number of still unanswered questions about neutrinos, that the NOvA experiment could shed light on.

The structure of this section is heavily inspired by the Steve Boyd's Lecture Notes on Neutrino Oscillations[1].

2.1 Physics of Neutrino Oscillations

I will begin the discussion of the neutrino physics with a quick reminder of the key facts about neutrinos from the Standard Model. According to the Standard Model, neutrinos are spin 1/2 particles that do not participate in the strong and electromagnetic interactions after the electroweak symmetry breaking. There are three known neutrino flavors corresponding to the three charged lepton generations (e, μ, τ).

Neutrinos do participate in the Weak interactions, and there are two distinctive types of neutrino interaction vertices that couple them to the weak gauge bosons – the

[1] https://web.archive.org/web/20200921172706/https://warwick.ac.uk/fac/sci/physics/staff/academic/boyd/stuff/neutrinolectures/lec_oscillations.pdf

Charged Current interaction vertex and the Neutral Current interaction vertex [4].

The Neutral Current neutrino interaction vertex describes the coupling of the neutrino current to the neutral vector boson Z^0_μ (Figure 2.1). This interaction vertex is diagonal in the neutrino flavors (i.e. it does not mix neutrinos between different lepton generations). The Charged Current neutrino interaction vertex couples charged lepton current to the charged vector bosons $W^\pm_\mu$ (Figure 2.1). Similar to the Neutral Current, the Charged Current interaction vertex is also diagonal in the lepton generations.

(a) Neutral Current Interaction Vertex (b) Charged Current Interaction Vertex

Figure 2.1: Neutral and Charged Current neutrino interaction vertices according to the Standard Model. l_α are the charged leptons ($\alpha \in \{e, \mu, \tau\}$) and ν_α are the corresponding neutrinos.

Until the discovery of the neutrino oscillations, the Standard Model assumed that the kinetic part of the Lagrangian $\mathcal{L}_0$ (that describes the neutrino propagation through the spacetime) is diagonal in the neutrino flavors as well. It was also assumed that neutrinos are massless particles. These assumptions allowed massless neutrino eigenstates ν_i (where $i \in \{1, 2, 3\}$) of the kinetic part of the Lagrangian $\mathcal{L}_0$ to be identified with the neutrino flavor states ν_α (where $\alpha \in \{e, \mu, \tau\}$).

The discovery of the neutrino oscillation phenomenon led physicists to revisit the basic assumptions about the physics of neutrinos. In particular, particle flavor oscillations can be observed if the free particle eigenstates ν_i of the kinetic part of the Lagrangian are non-degenerate (as is the case of different masses) and the particle flavor states ν_α are nontrivial combinations (i.e. $\nu_\alpha = \sum_i C_i \nu_i$ with nonzero C_i) of the free particle eigenstates.

To show mathematically the neutrino oscillation mechanism a framework of

Quantum Mechanics is usually employed, although more fundamental Quantum Field Theory treatments are also possible [5]. In the following sections, I will show how neutrino oscillations can be obtained in the Quantum Mechanical setting and derive a number of important equations characterizing this process.

2.1.1 Formalism

In order to derive mathematically the phenomenon of the neutrino oscillation, I will make a number of assumptions that are commonly made when studying it. First, I will assume that there are several free neutrino states ν_i that have unequal masses (m_i). Next, I will assume that neutrinos participate in the Weak interactions in flavor states ν_α, and that the flavor states are nontrivial combinations of the free neutrino states ν_i. The transformation between the basis of free states ν_i into the basis of flavor states ν_α is achieved by a unitary matrix $U_{\alpha i}$ which is also known as the neutrino mixing matrix (or PMNS matrix).

Let say, that neutrinos are produced at the source in a flavor state $|\nu_\alpha(t_{\text{src}}, x_{\text{src}})\rangle$ via some Weak interaction process. Then, they travel through spacetime from the source to the detector. Finally, we observe neutrinos at the detector in a flavor state $|\nu_\beta(t_{\text{det}}, x_{\text{det}})\rangle$ via some other Weak interaction process. The goal of the derivation of the neutrino oscillation mechanism is to calculate the probability of a neutrino being produced in a flavor α at the source and observed in a flavor β at the detector, i.e. we need to find $P(\nu_\alpha \to \nu_\beta) \equiv |\langle \nu_\beta(t_{\text{det}}, x_{\text{det}}) | \nu_\alpha(t_{\text{src}}, x_{\text{src}})\rangle|^2$.

A number of further assumptions can be made in order to simplify the calculation of the neutrino oscillation probability. Moreover, there are multiple different possible hypotheses under which such probability can be derived and all of them give the same answer [5]. For the sake of brevity, I will use one of such simplifying assumptions and assume that the flavor neutrinos ν_α have been produced at the source in the plain-wave state of fixed momentum $\vec{p}$ at a time t_{src} (2.1). I will also work in the natural units $\hbar \equiv c \equiv 1$ and then recover the proper dimensions when they are needed.

$$|\nu_\alpha(t_{\text{src}})\rangle = \sum_i U_{\alpha i} \, |\nu_i(\vec{p}, t_{\text{src}})\rangle \tag{2.1}$$

During the travel from the source to the detector, neutrino states evolve under the

action of the free Hamiltonian $\mathcal{H}_0$. Applying the time evolution operator $\exp(-i\mathcal{H}_0 t)$ to Equation 2.1 and using the fact that all neutrino states share the same momentum we get the neutrino state at moment t_{det}:

$$|\nu_\alpha(t_{\text{det}})\rangle = \sum_i U_{\alpha i} \exp\left(-iE_i(t_{\text{det}} - t_{\text{src}})\right) |\nu_i(\vec{p}, t = 0)\rangle \tag{2.2}$$

where $E_i = \sqrt{\vec{p}^2 + m_i^2}$ is the energy of a free neutrino state i and m_i is its mass. Due to the simplifying plain-wave hypothesis about the initial neutrino state, state (2.2) has uniform support over the entire space. In a realistic case, however, we would have a wave packet of neutrinos localized in space and traveling with a speed $v = p/E$, where $p = \sum_i p_i/n$ and $E = \sum_i E_i/n$ are the average energy and momentum of the wave packet (the critical assumption here is that the wave packet does not have enough time to dissipate into separate components while traveling from the source to the destination).

Likewise, in a realistic case neutrinos will be detected at some distance from the source L. Therefore, transitioning into a more realistic case of neutrino wave-packets I will work with distances instead of times and replace the time difference between the source and the detector by L/v (assuming neutrinos travel directly from the source to the destination):

$$|\nu_\alpha(x = L)\rangle = \sum_i U_{\alpha i} \exp\left(-i\frac{E_i}{p}\frac{E}{p}L\right) |\nu_i(\vec{p}, x = 0)\rangle \tag{2.3}$$

The scalar product of $\langle\nu_\beta(t_{\text{det}}, x_{\text{det}})|\nu_\alpha(t_{\text{src}}, x_{\text{src}})\rangle$, required to calculate the oscillation probability, will be given then by Equation 2.4.

$$\langle\nu_\beta(x = L)|\nu_\alpha(x = 0)\rangle = \sum_i U^*_{\beta i} U_{\alpha i} \exp\left(i\frac{E_i}{p}\frac{E}{p}L\right) \tag{2.4}$$

And the desired neutrino oscillation probability is

$$\begin{aligned} P\left(\nu_\alpha \to \nu_\beta\right) &= \left|\langle\nu_\beta(x = L)|\nu_\alpha(x = 0)\rangle\right|^2 \\ &= \sum_{i,j} U^*_{\beta i} U_{\beta j} U^*_{\alpha j} U_{\alpha i} \exp\left(i\left(E_i - E_j\right)\frac{EL}{p}\right) \end{aligned} \tag{2.5}$$

In the neutrino physics, it is further assumed that the masses of neutrinos are very small compared to their energies $m_i \ll E_i$ (ultra-relativistic limit). The ultra-relativistic limit allows us to make the following approximation $E_i \approx p\left(1+\frac{m_i^2}{2p^2}\right)$ (obtained using Taylor expansion of $E=\sqrt{p^2+m^2}$). Exploiting this assumption Equation 2.5 can be further simplified to Equation 2.6.

$$P\left(\nu_\alpha \to \nu_\beta\right) \approx \sum_{i,j} U^*_{\beta i} U_{\beta j} U^*_{\alpha j} U_{\alpha i} \exp\left(i\left(m_i^2 - m_j^2\right)\frac{EL}{2p^2}\right) \tag{2.6}$$

Finally, we can use the ultra-relativistic approximation once again, and keep only the leading terms in m/p in the arguments of exponents in Equation 2.6. In this limit, we can equate $p \approx E$, and obtain the final expression of the neutrino oscillation probability Equation 2.7.

$$P\left(\nu_\alpha \to \nu_\beta\right) = \sum_{i,j} U^*_{\beta i} U_{\beta j} U^*_{\alpha j} U_{\alpha i} \exp\left(i\Delta m_{ij}^2 \frac{L}{2E}\right) \tag{2.7}$$

where I have introduced a new parameter $\Delta m^2_{ij} \equiv m_i^2 - m_j^2$ – the squared difference of the neutrino masses. This parameter controls the neutrino oscillation probability and it is one of the most important parameters of the neutrino oscillation physics.

2.1.2 Two Flavor Neutrino Oscillation Case

The general expression (2.7) is too complex to be easily understood, therefore I will briefly consider a case of a two flavor neutrino oscillation in order to develop some intuition. Then, I will move to a more realistic case of a three flavor neutrino oscillation. In a two flavor neutrino oscillation case, the most general form of the neutrino mixing matrix is given by Equation 2.8.

$$U_{\alpha i} = \begin{pmatrix} e^{i\delta}\cos\theta & e^{i\epsilon}\sin\theta \\ -e^{-i\epsilon}\sin\theta & e^{-i\delta}\cos\theta \end{pmatrix} \tag{2.8}$$

The probability (2.7) of neutrino born in state α to be detected in state α at the

detector can be easily expanded in the two flavor case giving Equation 2.9.

$$P\left(\nu_\alpha \to \nu_\alpha\right) = 1 - 4\left|U_{\alpha 1}\right|^2\left|U_{\alpha 2}\right|^2\ \sin^2\left(\Delta m^2_{ij}\frac{L}{4E}\right) \tag{2.9}$$

Substituting here the mixing matrix (2.8) we will obtain the two flavor neutrino survival probability Equation 2.10. Using this probability we can also calculate the probability of a neutrino born in flavor α to oscillate into a different flavor β (2.11).

$$P\left(\nu_\alpha \to \nu_\alpha\right) = 1 - \sin^2\left(2\theta\right)\ \sin^2\left(\Delta m^2\frac{L}{4E}\right) \tag{2.10}$$

$$P\left(\nu_\alpha \to \nu_\beta\right) = \sin^2\left(2\theta\right)\ \sin^2\left(\Delta m^2\frac{L}{4E}\right) \tag{2.11}$$

These neutrino oscillation probabilities depend on two parameters. The first parameter is the neutrino mixing angle θ. If there is no mixing between neutrino flavors (i.e. $\theta = 0 + n\pi/2$) then the neutrino born in flavor α will always stay in flavor α. On the other hand, the maximum of the neutrino oscillation probability (2.11) will be achieved when the neutrino mixing angle $\theta = \pi/4 + n\pi/2$.

The neutrino mixing angle determines the magnitude of the neutrino oscillation probability. However, the neutrino oscillation probability (2.11) is further modulated by a periodic function of the argument $\phi \equiv \left(\Delta m^2 \frac{L}{4E}\right)$. Therefore, when the mixing is maximal, along the trajectory of a beam of neutrinos produced in flavor α, there will be places where the neutrinos will interact as if the beam is fully made of neutrinos in a different flavor β (e.g. when $\phi = \pi/2 + n\pi$). Naturally, there will also be places (e.g. when $\phi = 0 + n\pi$) where the beam will interact in the original fashion. And between the two extremes $\phi = 0$ and $\phi = \pi/2$ the probability of neutrino oscillating into a different flavor will be modulated by a $\sin^2\phi$.

Finally, the neutrino oscillation probability (2.11) is sensitive only to the absolute value of the squared mass difference Δm^2 and it is not sensitive to its sign. This observation will become important later.

2.1.3 Three Flavor Case

After we have briefly looked at a toy case of the two flavor neutrino oscillation I will move on to the realistic case of the three flavor neutrino oscillations. In the case of three neutrino flavors, the neutrino mixing matrix $U_{\alpha i}$ (PMNS matrix) is usually parametrized as a matrix product (2.12) [6].

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{23} & \sin\theta_{23} \\ 0 & -\sin\theta_{23} & \cos\theta_{23} \end{pmatrix} \begin{pmatrix} \cos\theta_{13} & 0 & \sin\theta_{13}e^{-i\delta_{\mathrm{CP}}} \\ 0 & 1 & 0 \\ -\sin\theta_{13}e^{i\delta_{\mathrm{CP}}} & 0 & \cos\theta_{13} \end{pmatrix} \begin{pmatrix} \cos\theta_{12} & \sin\theta_{12} & 0 \\ -\sin\theta_{12} & \cos\theta_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.12}$$

where θ_{12}, θ_{23}, θ_{13} are the neutrino mixing angles and δ_{CP} is the CP violation phase. Similar to the case of the two flavor neutrino oscillation, the neutrino mixing angles determine the magnitude of the neutrino oscillations. Unlike the two flavor case, however, the three flavor case also depends on the CP violation phase that cannot be absorbed by a redefinition of the neutrino fields. This phase controls the magnitude of the CP symmetry violation in the lepton sector of the Standard Model.

To simplify the analysis of the neutrino oscillation probabilities I will note that the probability defined by Equation 2.7 is real. Therefore, using unitarity of the mixing matrix $U_{\alpha i}$, it can be rewritten as (2.13).

$$P\left(\nu_\alpha \to \nu_\beta\right) = \delta_{\alpha\beta} - 4\sum_{i>j} \mathrm{Re}\left[U^{ij}_{\alpha\beta}\right] \sin^2\left(\Delta m^2_{ij}\frac{L}{4E}\right) - 2\sum_{i>j} \mathrm{Im}\left[U^{ij}_{\alpha\beta}\right] \sin\left(\Delta m^2_{ij}\frac{L}{2E}\right) \tag{2.13}$$

where I have introduced the notation $U^{ij}_{\alpha\beta} \equiv U^*_{\beta i}U_{\beta j}U^*_{\alpha j}U_{\alpha i}$. A number of interesting observations can be drawn from Equation 2.13. When the CP symmetry violation is absent (i.e. δ_{CP} takes values $\delta_{\mathrm{CP}} = \pi N$ for some integer N) the imaginary part of Equation 2.13 vanishes and it reduces to the form similar to the two flavor neutrino oscillations (2.9). Therefore, in the absence of the CP violation, the results of the two flavor neutrino oscillation analysis can be applied to the three flavor oscillations.

The presence of the CP symmetry violation has some consequences for neutrino oscillation probability. First, the imaginary term of Equation 2.13 is sensitive to the sign of the Δm^2_{ij}, unlike the real term. In other words, in the presence of the CP violation, the neutrino oscillation probability is affected by the sign of Δm^2_{ij}. Second,

since antineutrinos are described by conjugate fields, the mixing matrix for antineutrinos will be $U^*_{\alpha i}$ and the oscillation probability for antineutrinos $P\left(\bar{\nu}_\alpha \to \bar{\nu}_\beta\right)$ will be determined by the same expression as Equation 2.13, except that the mixing matrix will be replaced by its conjugate. The presence of the imaginary term in Equation 2.13 creates a nonzero difference between the oscillation probabilities for neutrinos and antineutrinos (2.14). The difference between the oscillation probabilities for neutrinos and antineutrinos is one of the manifestations of the CP symmetry violation.

$$P\left(\nu_\alpha \to \nu_\beta\right) - P\left(\bar{\nu}_\alpha \to \bar{\nu}_\beta\right) = -4 \sum_{i>j} \mathrm{Im}\left[U^{ij}_{\alpha\beta}\right] \sin\left(\Delta m^2_{ij} \frac{L}{2E}\right) \tag{2.14}$$

One can substitute the PMNS matrix directly into Equation 2.13 in order to obtain the final expression for the neutrino oscillation probabilities. However, the resulting expression will be rather complex and difficult to analyze analytically. Therefore, instead of analyzing the most general case of neutrino oscillation probability (2.13) with three neutrino flavors, I will consider a number of edge cases that are relevant for the experimental neutrino field.

Experimental Approximations

When discussing cases relevant for the neutrino experiments it is useful to revisit the known values of the neutrino oscillation parameters, and in particular the values of Δm^2_{ij}. From the solar and atmospheric neutrino oscillation experiments, we know that $\Delta m^2_{21} = \mathcal{O}\left(10^{-5}\mathrm{eV}^2\right)$ and $|\Delta m^2_{32}| = \mathcal{O}\left(10^{-3}\mathrm{eV}^2\right)$. The value of the Δm^2_{31} is constrained by the identity $\Delta m^2_{21} + \Delta m^2_{32} + \Delta m^2_{13} \equiv 0$. Comparing values of the squared mass differences we can see that $|\Delta m^2_{32}| \gg |\Delta m^2_{21}|$. Therefore, via the mass identity above we get another relation $|\Delta m^2_{31}| \approx |\Delta m^2_{32}|$.

The neutrino oscillation parameters Δm^2_{21} enter the expression for the neutrino oscillation probability (2.7) in arguments of the trigonometric functions of the form $(\Delta m^2_{21} L/E)$. This fact allows us to investigate separate cases of the neutrino oscillation probability (2.7) depending on whether the ratio of L/E is "large" or "small" compared to the Δm^{-2}_{21}.

2.1.4 Three Flavor Neutrino Oscillations for Small L/E

For the neutrino experiments where $\Delta m^2_{21} L/E \ll 1$ while $\Delta m^2_{32} L/E \sim 1$, Equation 2.7 can be simplified by neglecting the first and higher-order terms in $\Delta m^2_{21} L/E$. Using further approximation of $|\Delta m^2_{31}| \approx |\Delta m^2_{32}|$, Equation 2.7 becomes (2.15).

$$\begin{aligned} P(\nu_\alpha \to \nu_\beta) \approx \delta_{\alpha\beta} - 4\mathrm{Re}\left[U^{32}_{\alpha\beta} + U^{31}_{\alpha\beta}\right] \sin^2\left(\Delta m^2_{32} \frac{L}{4E}\right) \\ - 2\mathrm{Im}\left[U^{32}_{\alpha\beta} - U^{31}_{\alpha\beta}\right] \sin\left(\Delta m^2_{32} \frac{L}{2E}\right) \end{aligned} \tag{2.15}$$

The neutrino oscillation probability (2.15) for small ratios of L/E is sensitive mostly to the squared mass difference Δm^2_{32} (equal to Δm^2_{31} by our approximation). Additionally, if one also assumes that $\theta_{13} \ll 1$ (experimentally $\theta_{13} \approx 9\,\mathrm{deg}$) and only considers the 0th order terms in θ_{13} then the majority of the probabilities (2.15) will become trivial. The only nontrivial remaining oscillation probability will describe mixing between μ and τ sectors (2.16).

$$P(\nu_\mu \to \nu_\tau) \approx \sin^2(2\theta_{23}) \sin^2\left(\Delta m^2_{32} \frac{L}{4E}\right) \tag{2.16}$$

Equation 2.16 shows that the experiments with low ratios of L/E are mostly sensitive to the mixing angle θ_{23} and the squared mass difference Δm^2_{32}. Therefore, such experiments can be used to make precision measurements in the $2-3$ sector.

2.1.5 Three Flavor Neutrino Oscillations for Large L/E

Another neutrino oscillation case that is relevant for the neutrino oscillation experimental field is the case when the ratio of L/E is large compared to $1/\Delta m^2_{32}$, i.e. when $\Delta m^2_{32} L/E \gg 1$. For example, this can happen for neutrinos with small energies. In such a case, small changes of L (as neutrino moves within the volume of the detector) will make the argument of the trigonometric functions in Equation 2.13 run through multiple numbers of periods of this function. In other words, the trigonometric functions in Equation 2.13 that depend on the parameter $\Delta m^2_{32} L/E$ will oscillate so fast in the detector volume, that we will be able to see only the average picture produced

by such oscillations. Averaging such oscillatory terms of Equation 2.13 and assuming that $\Delta m^2_{21} L/E \sim 1$ one can obtain Equation 2.17.

$$\begin{aligned} P\left(\nu_\alpha \to \nu_\beta\right) \approx \delta_{\alpha\beta} - 2\mathrm{Re}\left[U^{32}_{\alpha\beta} + U^{31}_{\alpha\beta}\right] - 4\mathrm{Re}\left[U^{21}_{\alpha\beta}\right] \sin^2\left(\Delta m^2_{21}\frac{L}{4E}\right) \\ - 2\mathrm{Im}\left[U^{21}_{\alpha\beta}\right] \sin\left(\Delta m^2_{21}\frac{L}{2E}\right) \end{aligned} \tag{2.17}$$

In the limit of large L/E, the neutrino oscillation probability is mostly sensitive to the Δm^2_{21} and not sensitive to the Δm^2_{32} (and Δm^2_{31}). Similar to the case of the small L/E we can use an additional approximation of $\theta_{13} \ll 1$ in order to further simplify the neutrino oscillation probabilities (2.18).

$$\begin{aligned} P\left(\nu_e \to \nu_\mu\right) \approx P\left(\nu_\mu \to \nu_e\right) \approx \cos^2\left(\theta_{23}\right) \sin^2\left(2\theta_{12}\right) \sin^2\left(\Delta m^2_{21}\frac{L}{4E}\right) \\ P\left(\nu_e \to \nu_\tau\right) \approx P\left(\nu_\tau \to \nu_e\right) \approx \sin^2\left(\theta_{23}\right) \sin^2\left(2\theta_{12}\right) \sin^2\left(\Delta m^2_{21}\frac{L}{4E}\right) \end{aligned} \tag{2.18}$$

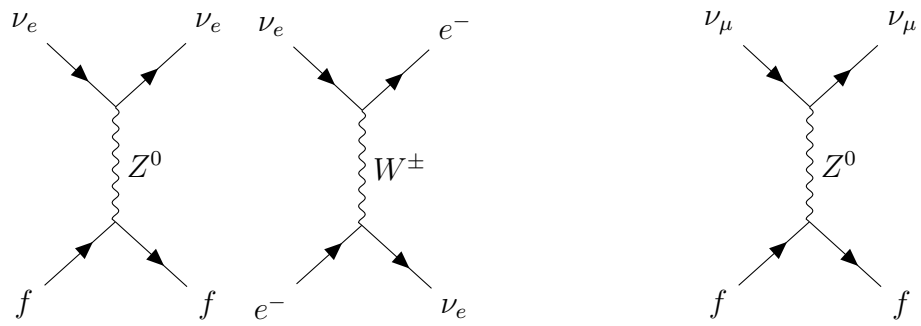
In the limit of small θ_{13}, the probability of an electron neutrino ν_e to oscillate into the other flavors $P\left(\nu_e \to \nu_\mu + \nu_\tau\right)$ is dependent only on the θ_{12} and Δm^2_{21}. Therefore, neutrino experiments having large ratios of L/E can be used to make precision measurements in the $1-2$ sector.

2.1.6 Neutrino Oscillations in Matter

So far I have considered the neutrino oscillation phenomenon as it would happen in a physical vacuum. However, in reality, neutrinos usually travel through some amount of matter before reaching the detector. While traveling through this matter neutrinos interact weakly with it. The interaction of neutrinos with matter depends on the flavor of neutrinos [6]. And this dependence on the flavor modifies the neutrino oscillation probabilities.

The dependence of neutrino interaction probabilities with matter on the neutrino flavor stems from the fact, that the ordinary matter is made of protons, neutrons, and electrons. All neutrino flavors can interact elastically with matter via the neutral current interactions. However, only electron neutrinos ν_e can elastically scatter off

electrons in the atomic orbitals via the charged current interaction (Figure 2.2).



(a) ν_e interactions with ordinary matter (b) ν_μ interactions with ordinary matter

Figure 2.2: Possible elastic interactions of the ν_e and ν_μ with the ordinary matter. f denotes a fermion (p, n, e).

Since the interactions of neutrinos with matter are weak, the effective interaction Hamiltonian is diagonal in the neutrino flavor basis $\mathcal{H}^M_{\text{int}} = \sum_\alpha V_\alpha |\nu_\alpha\rangle \langle\nu_\alpha|$ (V_α depends on a density of electrons and on the neutrino scattering cross-section $S \sim G_F E$). This in turn means that if neutrino in a free mass state $|\nu_i\rangle$ is traveling through matter, the interaction Hamiltonian $\mathcal{H}_{\text{int}}$ will make it mix with the other free mass states $|\nu_j\rangle$. The mixing of the free neutrino states in the presence of matter requires us to find a new set of states $|\nu^M_j\rangle$ in order to describe the neutrino propagation through matter.

One can find a new basis of neutrino states $|\nu^M_j\rangle$ by diagonalizing the kinetic part of the Hamiltonian in the presence of matter $\mathcal{H}_0 + \mathcal{H}^M_{\text{int}}$. Then, the neutrino oscillation analysis will have to be repeated, but now using the new "free" states $|\nu^M_j\rangle$ instead of the vacuum states $|\nu_j\rangle$. The result of this analysis will be an equation similar to Equation 2.7, but with the neutrino oscillation parameters renormalized due to interactions with matter. It is instructive to look at the formula for the neutrino oscillation probability for the case of two neutrino flavors in the presence of matter [6].

$$P^M(\nu_e \to \nu_\mu) = \sin^2(2\theta_M) \sin^2\left(\Delta m^2_M \frac{L}{4E}\right) \tag{2.19}$$

The neutrino oscillation probability formula in the presence of matter (2.19) has the same form as in a vacuum (2.11), but with renormalized mixing angle and squared

mass difference (2.20).

$$\begin{aligned} \Delta m^2_M &= \Delta m^2 \sqrt{\left(\Delta V/\Delta m^2 - \cos\left(2\theta\right)\right)^2 + \sin^2\left(2\theta\right)} \\ \sin\left(2\theta_M\right) &= \frac{\sin\left(2\theta\right)}{\sqrt{\left(\Delta V/\Delta m^2 - \cos\left(2\theta\right)\right)^2 + \sin^2\left(2\theta\right)}} \end{aligned} \tag{2.20}$$

where ΔV is the difference between the effective interaction potentials for muon and electron neutrinos. The presence of matter produces a number of interesting effects on the neutrino oscillation parameters:

- The neutrino oscillation parameters in matter are sensitive to the sign of Δm^2. This makes the neutrino oscillation probability in matter (2.19) sensitive to the sign of Δm^2 as well. Therefore, it might be possible to resolve the sign of the squared mass difference in the neutrino oscillation experiments, if these experiments have a significant matter effect.

- If matter density is large $\Delta V/\Delta m^2 \gg 1$ then it will significantly suppress neutrino oscillations.

- On the other hand, there are some values of matter density where

  $$\left(\Delta V/\Delta m^2 - \cos\left(2\theta\right)\right)^2 = 0$$

  In such regimes, the effective neutrino mixing angle will be maximal $\sin\left(2\theta_M\right) = 1$, irrespective of how large the original mixing was. This effect is called MSW resonance [6].

2.2 Neutrino Oscillation Experiments

Historically, there have been several epochs of neutrino oscillation experiments. The earliest neutrino oscillation experiments were aimed at simply capturing neutrinos originating from natural sources and determining whether the neutrino oscillation phenomenon exists. Once neutrino oscillations have been discovered a new generation of neutrino experiments was constructed. These experiments use artificial

neutrino sources and aim to verify the results of the first generation experiments and get rough estimates of the neutrino oscillation parameters.

At the present moment, the physical model of the neutrino oscillations has been experimentally confirmed and rough measurements of the neutrino oscillation parameters have been obtained. The next generation of the neutrino oscillation experiments is being constructed that will work with high intensity neutrino beams and will try to get precise measurements of the neutrino oscillation parameters.

The content of this section will heavily rely on the material from [6].

2.2.1 Discovery of the Neutrino Oscillations

The Solar Neutrino Problem

In the 1960s a model of the Sun's interior was created. This model allowed physicists to make predictions about the flux of electron neutrinos produced in fusion reactions inside of the Sun. A number of experiments were built in order to detect electron neutrinos coming from the Sun and confirm the Solar Model.

The first experiment that tried to measure the solar neutrino flux was the Homestake experiment (1965) [6]. The Homestake experiment used a large reservoir filled with tetrachloroethene $C_2\,Cl_4$. The electron neutrino coming from the Sun can interact with chlorine isotope $^{37}\mathrm{Cl}$ producing an argon atom: $^{37}\mathrm{Cl} + \nu_e \rightarrow {}^{37}\mathrm{Ar} + \mathrm{e}^-$. From time to time, the argon gas was scrubbed from the reservoir and the number of argon atoms in it was estimated. The estimate of argon atoms was used to get an estimate of the number of solar neutrino interactions in the reservoir. After several years of running, it was determined that the measured number of neutrino events was three times smaller than the number of events predicted by the Solar Model [6].

Such a large discrepancy between the measured and predicted number of neutrino events was a cause for concern and was called the Solar Neutrino Problem. However, the Homestake was the first solar neutrino experiment and no more data about the solar neutrinos was available. Therefore, it was possible that the Homestake experiment is simply wrong. Another possibility was that the Solar Model is incorrect. More observations were required to rule out these possibilities.

The Super-Kamiokande experiment provided another handle to solve this prob-

lem. The SuperK experiment used a water Cherenkov detector in order to find neutrino interactions [6]. In water detectors, neutrinos can elastically scatter off electrons $\nu_e + e^- \to \nu_e + e^-$ (2.2) and if the final electron has energy above 5 MeV then it will produce Cherenkov radiation that can be observed in the detector.

The SuperK experiment had observed a deficit of the electron neutrinos compared to the Solar Model [6]. This observation strengthened the belief in the correctness of the Homestake results. In addition to that, the SuperK experiment was able to resolve the direction of the electron neutrinos and established that they were indeed coming from the Sun. That dispelled another possible doubt about the source of the neutrinos in the Homestake experiment.

A new set of experiments (SAGE, GALLEX, etc) sensitive to much less energetic neutrinos compared to Homestake and SuperK has also confirmed a deficit of electron neutrinos compared to the prediction of the Solar Model [6]. After these experiments, no doubt was left that the observed solar neutrino flux is significantly below the Solar Model prediction. On the other hand, the Solar Model itself was confirmed by various helioseismological studies. Therefore, it was believed that something happens to the electron neutrinos as they travel from the Sun to Earth. One of the proposals was that electron neutrinos oscillate to other neutrino flavors as they travel. To test the neutrino oscillation hypothesis a new experiment SNO was constructed.

The SNO (Sudbury Neutrino Observatory) experiment started taking data in 1999 and it used a heavy water Cherenkov detector to measure the total flux of all neutrino flavors coming from the Sun [6]. The deuterium atoms in heavy water are very fragile and can be easily broken by the neutral current interactions with neutrinos: $d + \nu_\alpha \to n + p + \nu_\alpha$. This interaction channel is sensitive to the combined flux of all neutrino flavors. The SNO experiment discovered that the combined neutrino flux coming from the Sun exactly matches the prediction of the Solar Model. However, the flux of electron neutrinos alone is significantly smaller than the prediction. This discovery showed that the electron neutrinos originating in the Sun oscillate to the other neutrino flavors as they travel to Earth.

The confirmation that the total neutrino flux from the Sun corresponds to the flux predicted by the Solar Model has resolved the Solar Neutrino Problem and definitely established presence of the neutrino oscillation phenomenon. However, while the

neutrino oscillation phenomenon was confirmed, the data from the solar neutrino experiments alone was not sufficient to validate the neutrino oscillation model that was presented in section 2.1.

Interpretation of the Solar Neutrino Data

At this moment we know that the neutrino oscillation model cannot be directly applied to the solar neutrino data. The first reason for that is that while traveling from the Sun to Earth neutrino wave-packets experience a large degree of decoherence. This decoherence makes most of the analysis of section 2.1 inapplicable. Another reason is that the electron neutrinos formed in the core of the Sun are surrounded by a matter with a very high density. The high density of the matter suppresses neutrino oscillation initially as was discussed in subsection 2.1.6. However, as neutrino travels outwards the density of matter decreases and eventually it may reach the value corresponding to the MSW resonance that is going to enhance the neutrino oscillation.

Careful analysis of the solar neutrino oscillations shows that the solar electron neutrinos ν_e oscillate predominantly into ν_μ [6]. It also provides rough estimates of the neutrino oscillation parameters $\Delta m^2_{21} \approx 6.6 \times 10^{-5}\text{eV}^2$ and $\tan^2\theta_{12} \approx 0.46$ [7].

Atmospheric Neutrino Anomaly

Another piece of the neutrino oscillation puzzle came from the observation of the neutrinos produced in the atmosphere. The Earth's atmosphere is constantly bombarded by high energy cosmic rays which are primarily made of high energy protons. When these high energy cosmic protons interact with the atmosphere they create cascades of secondary particles (called showers). The resulting showers are made of an entire zoo of various particle types, but in particular, they contain charged pions. The interesting thing about charged pions $\pi^\pm$ is that their predominant decay mode produces muon neutrinos:

$$\pi^+ \to \mu^+ + \nu_\mu, \quad \pi^- \to \mu^- + \bar{\nu}_\mu \tag{2.21}$$

Moreover, muons created in these decays undergo further decays, producing even

more neutrinos:

$$\mu^+ \rightarrow e^+ + \bar{\nu}_\mu + \nu_e, \quad \mu^- \rightarrow e^- + \nu_\mu + \bar{\nu}_e \tag{2.22}$$

The combined effect of two decays (2.21) and (2.22) is that the number of produced muon neutrinos is twice the number of produced electron neutrinos. And while there are other processes in the showers that produce neutrinos, the detailed calculations show that the fraction $R \equiv N_{\nu_\mu}/N_{\nu_e}$ is approximately equal to 2 [8].

Multiple experiments (SuperK, Soudan2, IMB, etc) have tried to observe the atmospheric neutrinos and verify the predicted ratio of ν_μ to ν_e [6]. However, all experiments observed a deficit of the ratio of ν_μ to ν_e compared to the prediction. The observed deficit was labeled the Atmospheric Neutrino Anomaly.

Apart from detecting the deficit of the muon neutrinos, the SuperK experiment was able to resolve the angle of the incoming neutrinos [8]. The SuperK results show that the flux of the atmospheric electron neutrinos ν_e is in a good agreement with the prediction. The flux of the atmospheric muon neutrinos agrees with predictions for the atmospheric neutrinos that are coming from the top of the detector. However, the deficit is observed for the muon neutrinos coming from the sides of the detector, and the deficit is very prominent for the neutrinos that are coming from the bottom of the detector [8].

The results of the SuperK experiment can be easily understood in the framework of the neutrino oscillations that I have presented in section 2.1. Since SuperK observed almost no changes in the ν_e flux compared to the prediction, but a large deficit of the ν_μ flux for the neutrinos coming from the bottom of the detector, it may be speculated that the reason for this deficit is that the muon neutrinos oscillated into ν_τ flavor while traveling through Earth. In terms of the neutrino oscillation model, it would imply a large mixing angle θ_{23} (as defined by Equation 2.12) between muon and tau sectors.

Assuming that the majority of the ν_μ oscillate into ν_τ the results of the SuperK experiment can be analyzed in the two flavor approximation for mixing between ν_μ and ν_τ. In this approximation, the neutrino oscillation probability $P(\nu_\mu \rightarrow \nu_\tau)$ would

be given by Equation 2.23 (c.f. subsection 2.1.2).

$$P\left(\nu_\mu \to \nu_\tau\right) = \sin^2\left(2\theta_{\text{atm}}\right)\ \sin^2\left(\frac{e \times 10^{-6}}{4\hbar c}\ \Delta m^2_{\text{atm}}\left[\text{eV}^2\right]\ \frac{L\left[\text{km}\right]}{E\left[\text{GeV}\right]}\right) \tag{2.23}$$

where I have recovered the scale factors missing when working in the natural units. The neutrino oscillation probability (2.23) depends on the length L between the source of neutrinos (upper atmosphere) and the detector. This length can be estimated in the SuperK experiment from the angle of the incoming neutrinos. The results show that Equation 2.23 does indeed model the observed deficit of the ν_μ events for different L with $\sin^2\left(2\theta_{\text{atm}}\right) \approx 1$ and $\Delta m^2_{\text{atm}} \approx 2 \times 10^{-3}\text{eV}$.

The SuperK experiment proved not only that the neutrino oscillation phenomenon exists, but also that the probability of the atmospheric neutrino oscillations could be reasonably modeled using the neutrino oscillation framework developed in section 2.1.

2.2.2 Verification of the Neutrino Oscillation Model

The first generation of the neutrino experiments has confirmed the existence of the neutrino oscillation. They have also shown that the neutrino oscillation framework can be applied to find the neutrino oscillation probability at least in some cases. However, due to a large number of uncertainties in the observable quantities and the lack of control over the neutrino sources, it was not possible to definitely conclude that the neutrino oscillation model of section 2.1 is valid.

The new epoch of the neutrino oscillation experiments began aimed at fully validating the neutrino oscillation model and getting accurate measurements of the neutrino oscillation parameters using controlled sources of neutrinos.

Measurements in the 2-3 Sector

From the SuperK experiment we know that $\Delta m^2_{32} \approx 2 \times 10^{-3}\text{eV}^2$. In order to make a neutrino oscillation experiment sensitive to Δm^2_{32} one can work with the beam of muon neutrinos in the limit of small L/E (c.f. subsection 2.1.4). The accelerator based experiments fall in this category, since they can produce neutrino beams of

large energies (of the order of 1 GeV). In the limit of small L/E the ν_μ oscillation probability will be given by Equation 2.16. Therefore, for the muon neutrinos with energies around 1 GeV, the neutrino oscillation probability (2.16) will achieve its first maximum at $L \approx 600\,\text{km}$.

The accelerator based neutrino oscillation experiments with baselines of 600 km are doable, but require beams of large intensity due to the beam divergence over large distances. Historically, there have been two influential accelerator based experiments studying muon neutrino beams in order to estimate mixing parameters in the $2-3$ sector – MINOS and T2K [6]. The MINOS experiment used a ν_μ beam produced at Fermilab with average energies of 3.0 GeV and a neutrino detector in Northern Minnesota with a baseline of $L = 732\,\text{km}$. The T2K experiment operated on a ν_μ beam of lower energies 0.6 GeV and a smaller baseline $L = 295\,\text{km}$.

Both MINOS and T2K experiments investigated the disappearance of muon neutrinos from the beams, as they traveled from the source to the detector, and were able to confirm the neutrino oscillation probability formula (2.16). Using accelerator based experiments we can obtain more precise measurements of the neutrino oscillation parameters θ_{23} and Δm^2_{32} [9]: $\sin^2\theta_{23} = 0.573 - 0.635$ and $\Delta m^2_{32} = (2.46 - 2.61) \times 10^{-3}\text{eV}^2$ (1σ intervals).

Measurements in the 1-2 Sector

The neutrino oscillation analysis of subsection 2.1.5 shows that in order to construct an experiment that is sensitive to the neutrino oscillation parameters in the $1-2$ sector one has to work in the limit of large L/E ratios. The accelerator based neutrino oscillation experiments are not acceptable since for their neutrino beams of energies of the order of 1.0 GeV, the required baseline would need to be larger than the size of the Earth. Therefore, the experiments aimed at measurements in the $1-2$ sector are operating on electron neutrinos produced in nuclear reactors, since such neutrinos have energies of the order of a few MeV.

The KamLAND experiment in Japan is one of such experiments [6]. It receives electron antineutrinos from about 50 different nuclear reactors within a radius of a thousand kilometers. By looking at the disappearance of the electron antineutrinos as they travel from the reactors, the KamLAND experiment was able to confirm

the prediction (2.18) of the neutrino oscillation framework. Combined with the solar neutrino data it obtained the best measurements of the neutrino oscillation parameters in the $1-2$ sector [10]: $\tan^2\theta_{12} \approx 0.44$ and $\Delta m^2_{21} \approx 7.5 \times 10^{-5}\text{eV}^2$.

Measurements in the 1-3 Sector

There are multiple ways one can measure neutrino oscillation parameters in the $1-3$ sector. For example, the ν_e appearance probability in the ν_μ beams does depend on the mixing parameters θ_{13}. However, that probability is dependent on a combination of various oscillation parameters from different neutrino sectors. The entanglement of the neutrino oscillation parameters makes it difficult to get precision measurements in the $1-3$ sector using the experiments discussed above.

In order to make direct measurements in the $1-3$ sector, one can try to place a neutrino detector at some precise distance L from the neutrino source, such that terms in Equation 2.13 dependent on Δm^2_{21} will vanish due to the oscillation minimum (i.e. take $L = \frac{4E}{\Delta m^2_{21}}\,\pi N$ where N is an integer number). The electron neutrino survival probability, in this case, will be given by Equation 2.24.

$$P\left(\nu_e \to \nu_e\right) = 1 - \sin^2\left(2\theta_{13}\right)\,\sin^2\left(\frac{e\times 10^{-6}}{4\hbar c}\,\Delta m^2_{32}\left[\text{eV}^2\right]\,\frac{L\left[\text{km}\right]}{E\left[\text{GeV}\right]}\right) \tag{2.24}$$

where I have used another approximation of $|\Delta m^2_{32}| \approx |\Delta m^2_{31}|$.

The Daya Bay experiment in China [11] is a neutrino oscillation experiment that tries to detect electron antineutrinos produced in nuclear reactors in order to estimate the θ_{13} neutrino oscillation parameter. The Daya Bay sees neutrinos produced in 6 nuclear reactors using 6 detectors at different baseline lengths. It observed a deficit of the electron antineutrinos that consistent with Equation 2.24, and it was able to get the world best measurement of the θ_{13} [11]: $\sin^2\left(2\theta_{13}\right) \approx 0.08$.

2.2.3 Next Generation Neutrino Experiments

At the end of the first generation of the neutrino oscillation experiments, the physical model of the neutrino oscillations has been experimentally verified, and rough measurements of the neutrino oscillation parameters were obtained. However, there

are still multiple questions that remain unanswered about the neutrino physics, like

- What is the value of the CP violation phase δ_{CP}.
- What is the sign of mass difference in the 2-3 sector Δm^2_{32}? (Neutrino Mass Hierarchy Problem)
- Current measurements indicate that the value of neutrino mixing angle θ_{23} is consistent with $\pi/4$, but it is unclear whether it is exactly $\pi/4$. If it is exactly $\pi/4$ then it may indicate presence of a symmetry and a possible handle to a new physics.

In order to find answers to these questions and get more accurate measurements of the neutrino oscillation parameters, a new generation of neutrino oscillation experiments was designed. These experiments rely on high intensity neutrino sources that allow them to collect much larger samples of neutrino interactions. Combined with improved analysis techniques the new generation of the neutrino oscillation experiments may be able to resolve the remaining ambiguities of the neutrino oscillation physics. The NOvA neutrino oscillation experiment is one of such experiments.

Chapter 3

The NOvA Experiment

NOvA (NuMI Off-Axis ν_e Appearance) is a long-baseline accelerator based neutrino oscillation experiment located in the United States of America [12]. It operates on a muon (anti-)neutrino NuMI beam produced at the Fermilab facility. The beam is directed northward and it travels through the Earth about 800 km until it exits the surface in Northern Minnesota. As the ν_μ neutrino beam travels through the Earth some neutrinos oscillate into the other flavor states (ν_e and ν_τ). The NOvA experiment tries to measure the neutrino beam composition when it exits the surface and to make inferences about the neutrino oscillation parameters based on these measurements.

The NOvA experiment follows the standard design of the long-baseline accelerator based experiments and has two main detectors – the Near and Far Detectors (Figure 3.1). The Near Detector is built close to the beam production site at Fermilab and it is used to measure the original composition of the beam. The Far Detector hall is built near the Ash River in Northern Minnesota. It is used to measure the composition of the oscillated neutrino beam.

The remainder of this chapter will be structured as follows. First, I will discuss the physical goals of the NOvA experiment. Then, I will describe in detail the design of the NOvA detectors and data acquisition system. Finally, I will conclude this chapter by summarizing the types of events that the NOvA detectors will be capable of distinguishing.

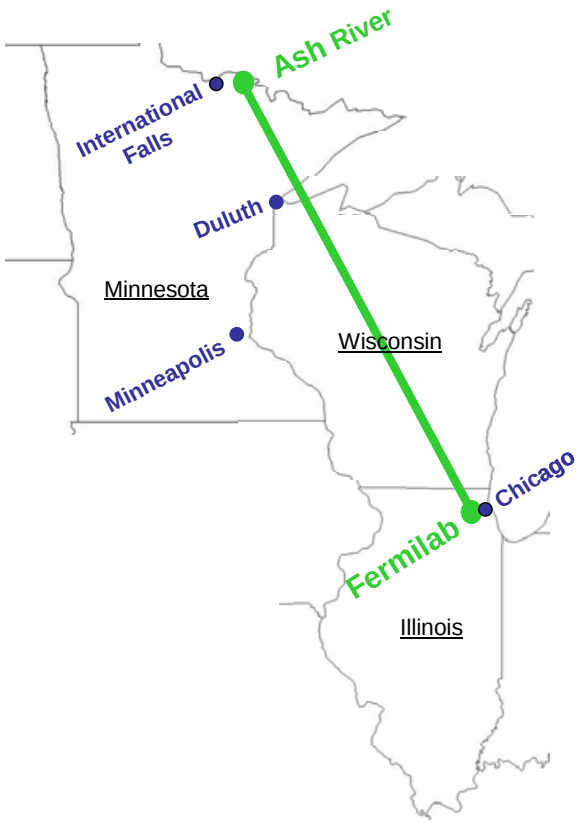


Figure 3.1: Locations of the NOvA Near and Far Detectors. Source [12].

3.1 Physical Goals of the NOvA Experiment

The NOvA experiment was primarily designed to observe the neutrino oscillation mode $\nu_\mu \rightarrow \nu_e$. Direct study of this oscillation mode will allow NOvA to refine constraints on multiple oscillation parameters. In particular, the NOvA experiment will likely be able to constrain the following parameters:

- Mass Splitting $|m^2_{32}|$. The NOvA experiment is expected to be highly sensitive to the value of the $|m^2_{32}|$, being an experiment with a small ratio of L/E (c.f. subsection 2.1.4).

- Mixing angle θ_{23} – current evidence indicates that the value of θ_{23} is consistent with the maximal mixing hypothesis ($\theta_{23} = \pi/4$). However, there is large uncertainty on the value. NOvA will be able to improve measurements of θ_{23}

and possibly resolve the question of whether the Maximal Mixing hypothesis is correct.

- Since the NuMI neutrino beam travels a significant distance through Earth it will be affected by the matter effects (c.f. subsection 2.1.6). The presence of the matter effects may allow NOvA to resolve the sign of m^2_{32}.

- Finally, NOvA is expected to run half of the time operating on muon neutrinos and half of the time on muon antineutrinos. Observing oscillations of both ν_μ and $\bar{\nu}_\mu$ may allow NOvA to put a constraint on the δ_{CP} CP violation phase.

Additionally, there are several secondary objectives that the NOvA experiment could achieve which go under the umbrella of "exotic" studies. These studies include, but are not limited to a search of magnetic monopole, supernova neutrino detection, search for neutrinos that coincide with the gravitational wave detection, etc. The "exotic" studies are highly domain specific and lie outside of the scope of this thesis.

3.2 Design of the NOvA Experiment

As I mentioned above, the NOvA experiment performs studies with the NuMI beam produced at Fermilab. The composition of the original unoscillated NuMI beam is measured at the Near Detector and then compared to the composition of the oscillated beam at the Far Detector. In this section, we will review the production of the NuMI beam and the design of the NOvA detectors.

3.2.1 The NuMI Beam

The muon (anti-) neutrino beam for NOvA is produced at the Fermilab National Accelerator Laboratory in Illinois, USA. The neutrino beam is formed by colliding high energy (120 GeV) protons with the graphite target [13]. These collisions create multiple charged particles and in particular charged pions. The charged pions are focused by two magnetic horns into the decay pipe (Figure 3.2).

While traveling in the decay pipe, charged pions undergo decays, primarily into lepton/neutrino pairs $\pi^+ \to \mu^+ + \nu_\mu$ and $\pi^- \to \mu^- + \bar{\nu}_\mu$. By controlling the direction

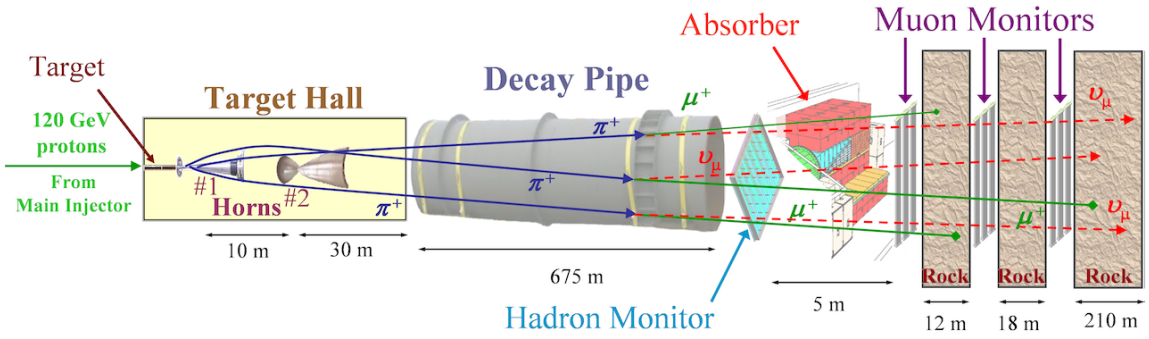


Figure 3.2: Schematic representation of the NuMI Beamline. Source [13].

of the current in the magnetic horns we can select either positively or negatively charged pions. This allows NOvA to operate in two modes. In the forward horn current (FHC) mode, positively charged pions are selected and the NuMI beam is made primarily of muon neutrinos. In the reverse horn current (RHC) mode, negatively charged pions are selected which makes the NuMI beam contain muon antineutrinos.

The hadron absorber is placed downstream of the decay pipe. Its purpose is to contain any remaining hadronic particles in the beam. Finally, the beam travels through roughly 250 m of rock, where the majority of muons are absorbed. The surviving particles in the beam are mostly muon (anti-) neutrinos. NOvA uses these (anti-) neutrinos in order to study the neutrino oscillation phenomenon.

The description above has left aside details about how the initial high energy protons are produced. The protons, required to create the NuMI beam, are accelerated by the Fermilab's synchrotron called the Main Injector (MI). Hence the name of the beam – NuMI – Neutrinos at the Main Injector. Several batches of protons are extracted from the Main Injector and shot onto the NuMI target giving a neutrino pulse of $8 - 10\,\mu$s (called spill window). After protons are extracted from the Main Injector, it takes some time to refill it and accelerate new protons to 120 GeV.

At the beginning of the NOvA experiment, it took about 2.2 s to recycle protons in the Main Injector [14]. However, multiple upgrades have been performed on the Main Injector which allowed us to reduce the interval between spills down to 1.3 s and simultaneously increase the beam's power. At the moment of writing, the NuMI

beam power was raised up to 700 kW (from 300 kW originally), which means that the MI delivers around 5.0×10^{13} protons on target (POT) in each spill.

3.2.2 NOvA Detectors

Both NOvA detectors are placed at around 14 mrad off the NuMI beam axis. The reason for such placement is tied to the way the NuMI beam is created. The neutrinos in the NuMI beam are the decay products of charged pions traveling along the decay pipe. Being decay products of the two particle decays $\pi^+ \to \mu^+ + \nu_\mu$ they have a broad distribution across angles and energies. However, for the purposes of the neutrino oscillation analysis, we are not interested in neutrinos of all energies, but in neutrinos with energies that maximize the oscillation probabilities (2.16). For the Far Detector, which is about $L = 810$ km from the NuMI target hall, the oscillation probability is maximized for the neutrinos of energies around 1.6 GeV.

Simulations show that when the detectors are placed at around 14 mrad off the NuMI beam axis, the neutrinos have a narrow neutrino energy distribution with a peak around 2 GeV (Figure 3.3).

The 14 mrad off the beam axis alignment was chosen since it provides many more neutrinos with optimal energies (1.6 GeV) for observing neutrino oscillations, and gives a beam with a narrower spectrum.

NOvA Detector Structure

Both NOvA detectors have an identical structure. They can be described as totally active, tracking, liquid scintillator calorimeters [13]. Each detector is made of a stack of planes along the beam direction (Figure 3.4). Each plane is composed of multiple cells – long PVC tubes filled with a liquid scintillator. In a given plane all cells are aligned in the same direction. However, the planes themselves are arranged in an alternating fashion, such that one plane has cells aligned horizontally, and the next one vertically.

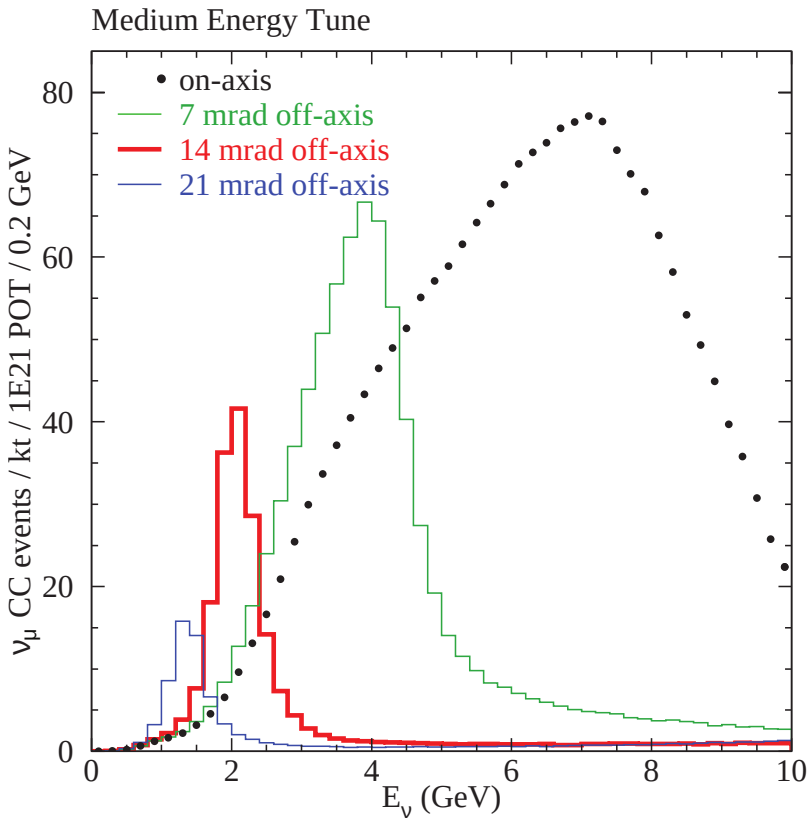


Figure 3.3: Simulated NuMI energy spectrum for different off-axis detector alignments. The NOvA detectors sit at around 14 mrad (red distribution). Source [13].

The NOvA Detector Cell

The basic unit of a NOvA detector is a rigid PVC cell. Each cell has an interior cross section of about (6 cm × 4 cm) in the directions along the beam and transverse to the beam respectively [15]. The Far Detector cells have a length of 15.6 m and the Near Detector cells are 4.1 m long. The walls of each cell are covered by titanium dioxide in order to increase their reflectivity.

The detector cells are filled with a liquid scintillator. The active scintillating agent is pseudocumene (1,2,4-Trimethylbenzene) which emits photons of $270 - 320$ nm wavelengths (UV range). The pseudocumene is dissolved in mineral oil and composes only 4.1% of the scintillator mass. The bulk of the remaining mass corresponds to the mineral oil with small amounts of

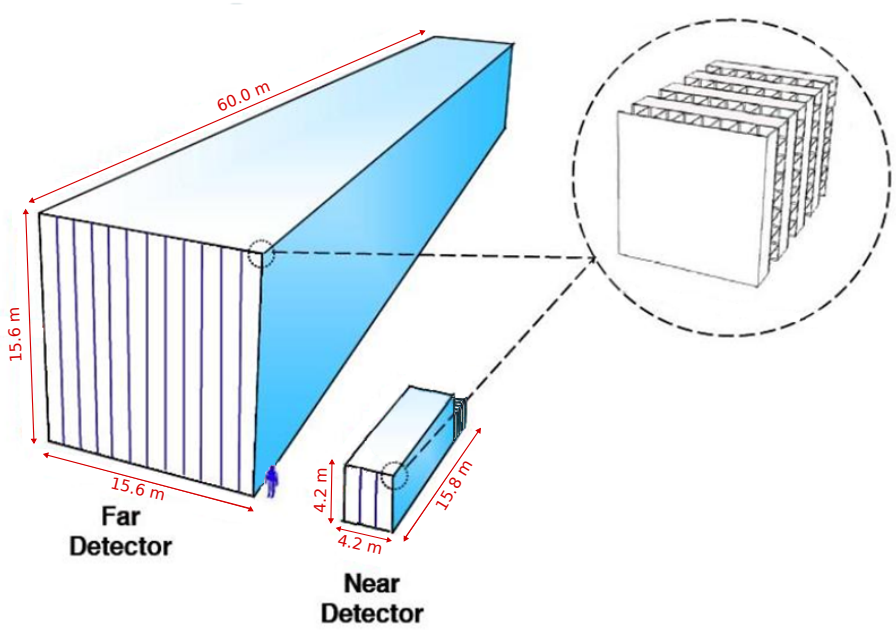


Figure 3.4: Schematic representation of the NOvA detectors.

- Wavelength shifting chemicals (PPO, bis-MSB) to shift the scintillator light into the $390 - 440\,\text{nm}$ range (blue range).
- Anti-Static agent (Stadius-425) to prevent build up of the hazardous amounts of charge in the cells.
- Antioxidant (Vitamin E) to prevent the scintillator degradation caused by oxygen exposure.

The light generated along the NOvA cell is collected by the wavelength shifting (WLS) fiber. The WLS fiber has a diameter of $0.07\,\text{mm}$ and forms a loop inside of each NOvA cell (Figure 3.5). The core of the fiber is made of polystyrene with trace amounts of fluorescent dye. The dye additive shifts light wavelengths from the blue part of the spectrum into the blue-green one ($450 - 650\,\text{nm}$). The short wavelength $< 520\,\text{nm}$ light is further attenuated as it travels through the fiber.

The ends of the fiber in each NOvA cell are connected to Avalanche Photodiode (APD). The APDs were chosen since they have a high 85% Quantum Efficiency for the $520 - 550\,\text{nm}$ light exiting the ends of the WLS fiber. To reduce thermal

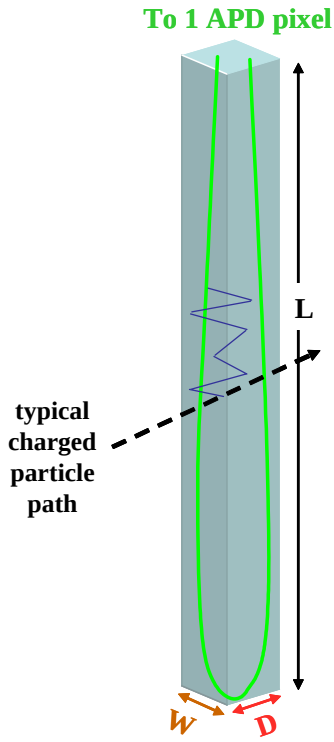


Figure 3.5: Schematic representation of the NOvA detector cell. Source [12].

noise generated by the APDs they are cooled down to $-15\,^{\circ}\mathrm{C}$ by the thermoelectric coolers.

The NOvA Cell Module

The detector cells are not produced individually, but instead in extrusions of 16 cells [12]. Two extrusions are glued together to form an extrusion module of 32 cells (Figure 3.6). One end of the cells in each extrusion module is covered by a simple plane of plastic to contain the liquid scintillator. The other end of the cells is outfitted with a complex fiber manifold, that routes 64 ends of the fibers (2 from each cell) to 32 pixels of the APD array.

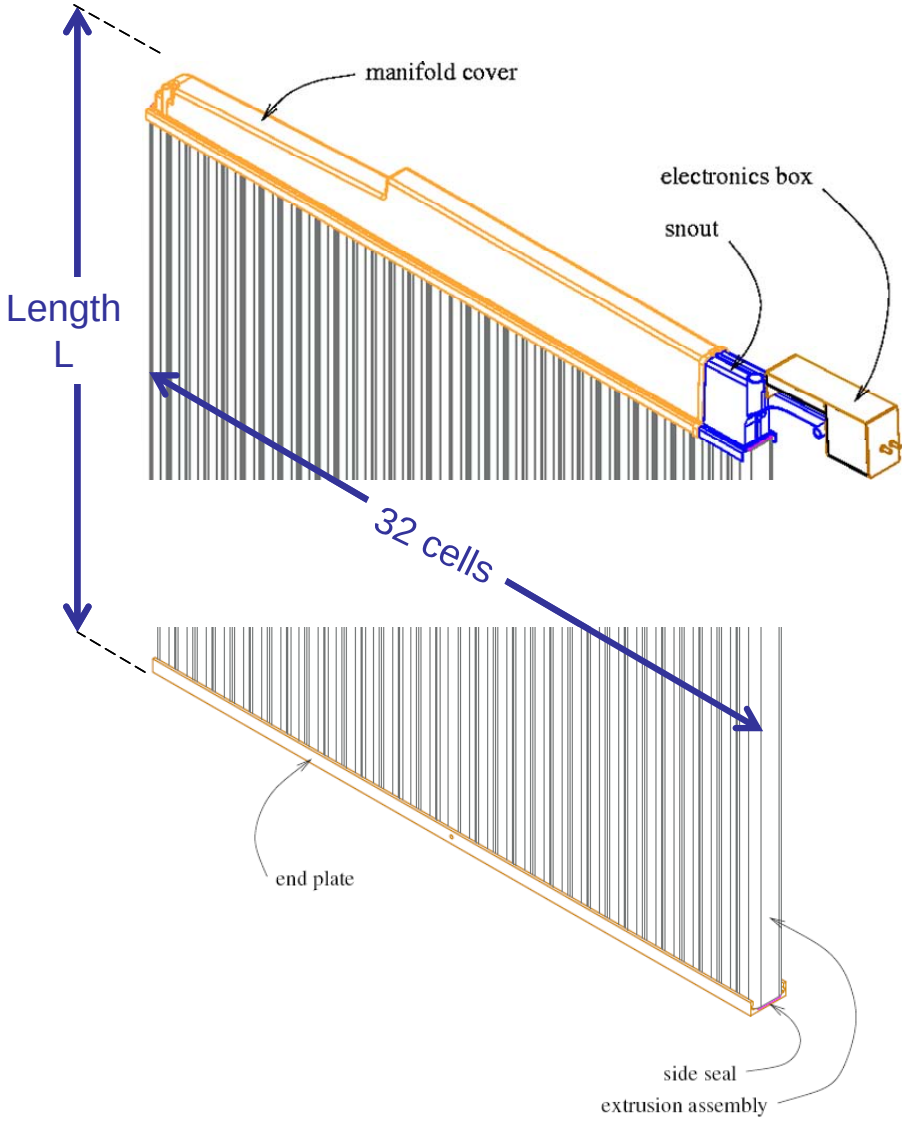


Figure 3.6: Schematic representation of the NOvA extrusion module. Source [12].

3.2.3 The NOvA Far Detector

The NOvA Far Detector hall is located at the Ash River site in Northern Minnesota, $L = 810.5\,\text{km}$ from the Fermilab and 14.6 mrad off the NuMI beam axis. The detector hall building is placed at the surface of Earth. Such placement, however, allows a large number of cosmic rays to enter the detector, creating significant cosmic backgrounds. To reduce the electromagnetic component of the cosmic backgrounds the detector is covered by a special shield of concrete and a loose barite material [12].

The NOvA Far Detector has dimensions of $(15.6 \times 15.6 \times 66.9)$ meters (width, height, length) and a mass of approximately 14 kt. It is made of 1003 alternating planes, where each plane is built of 12 extrusion modules and has 384 active channels [12]. The assembled planes are put against a concrete wall from one side and supported by a movable block pivoter from the other side (Figure 3.7). Usage of the movable block pivoter as support allows for adding additional detector planes in the future if such a need arises.

3.2.4 The NOvA Near Detector

The NOvA Near Detector is located at about 1015 m from the NuMI target hall, and 14.6 mrad off the beam axis. The Near Detector is 105 m below the surface level (Figure 3.8). At such depth, the earth above the detector provides effective shielding against the cosmic radiation.

Since the Near Detector is much closer to the target hall than the Far Detector (1 km vs 810 km) it receives a much higher neutrino flux. Correspondingly, less detector mass is required to detect neutrino interactions. The Near Detector has a mass of 222 ton, it is 2.8 m wide, 4.1 m tall, and has a length of 14.3 m along the beam direction.

However, the small length of the Near Detector creates its own problem. Muons produced in neutrino interactions have energies around $1 - 2\,\text{GeV}$, and they lose energy while traveling through the detector with a rate of about 1 GeV per 5 m traveled. This means that, on average, muons will travel $5 - 10\,\text{m}$ in the detector volume before losing all their kinetic energy. Therefore, a significant number of muons will escape the detector length (14.3 m). When muon escapes the detector

Figure 3.7: Photo of the Far Detector. The Far Detector side facing the reader is supported by a movable block pivoter (red).

volume it becomes very difficult to identify it and estimate its energy.

In order to work around the issue of escaping Near Detector muons, a 1.7 m long Muon Catcher is placed at the downstream end of the Near Detector. The Muon Catcher consists of 10 steel plates ($4.1 \times 2.9 \times 0.1\,\mathrm{m}^3$) interspersed by scintillator planes. The steel plates have a significant muon stopping power and are sufficient to stop most of the muons. The scintillator planes between them can be used to estimate distances that muons traveled in the Muon Catcher and make inferences about muon energies.

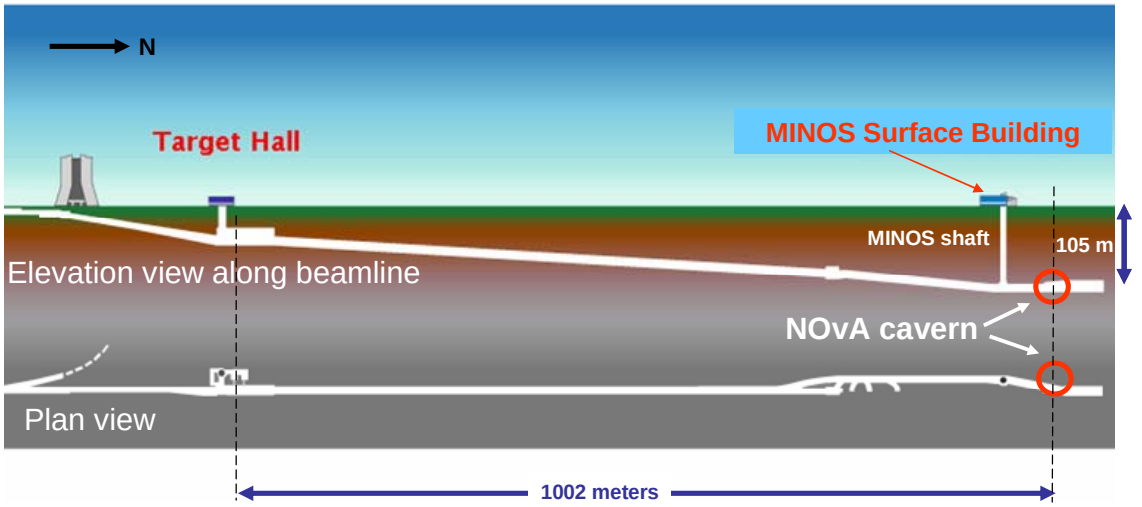


Figure 3.8: Plan view (bottom) and elevation view (top) of the NuMI beamline and the NOvA Near Detector. Source [12].

3.3 NOvA Data Acquisition System (DAQ)

As was discussed in the previous section, NOvA detectors are made of multiple cell modules, each made of 32 scintillator cells. The light produced in these cells is collected by optical fibers, which have ends that are attached to an array of APDs. In this section, I will discuss the organization of the NOvA data acquisition system responsible for the digitization of the raw detector data and preliminary data processing.

The APD array of the NOvA cell module is mounted on a Front-End Board (FEB). The FEBs continuously read data from the APDs, digitize it and send it to the Data Concentrator Modules (DCMs) along with timestamps over a Gigabit Ethernet network. A single DCM is responsible for accumulating data from 64 FEBs. The accumulated data in the DCM is sent to the farm of Buffer Nodes. Once the data reaches the farm of Buffer Nodes it is stored there for as long as the Buffer Node capacity allows, or until a decision is made whether to save or discard the data.

The decision on whether to save or discard the data is based on a system of triggers. The NOvA experiment relies on a variety of different triggers for detecting different event types. For example, the NuMI trigger is issued by Fermilab every

time a NuMI beam spill is generated. Once the Buffer Node farm receives the NuMI trigger it stores 500 µs of data, centered around the trigger timestamp (corrected for the time of flight of neutrinos), to the permanent location. If none of the triggers is activated for a given chunk of data then it is permanently discarded.

Special care is taken in order to synchronize time between all FEBs. Both detectors are equipped with two timing systems, one is primary and the other one serving as a backup. Each timing system has a master unit that is directly synchronized to the GPS clock. The GPS time is distributed among the DCMs taking into account various transmission delays in physical cables. Furthermore, each DCM, in turn, distributes the GPS time to the FEBs it is responsible for.

3.3.1 NOvA Event Displays

Since NOvA detectors are made of planes with alternating cell directions in them, the activity in the detector is usually represented as an event display with two views (c.f. Figure 3.9). The top view of the event display corresponds to the activity in the planes with vertically aligned cells. The bottom view shows activity in the planes with horizontally aligned cells.

Figure 3.9 can be used to illustrate the scope of difficulties that NOvA has to overcome to reconstruct neutrino interactions in the detector. First, Figure 3.9 contains a large amount of activity in the 500 µs window around the NuMI spill. Most of this activity is not related to the neutrino interactions from the NuMI beam and produced by cosmic muons. In order for NOvA to be able to analyze NuMI neutrinos, we have to filter out these cosmic muon backgrounds. Second, since NOvA detectors have two 2D views, we need to find a way to combine information from these two views to get a faithful representation of the 3D interaction. In chapter 5 I will discuss various approaches that NOvA uses in order to solve these difficulties.

3.4 Sensitivity of the NOvA Experiment

The NuMI neutrino beam that the NOvA experiment is using has a very high intensity. The high intensity of the neutrino beam will allow NOvA to collect a relatively

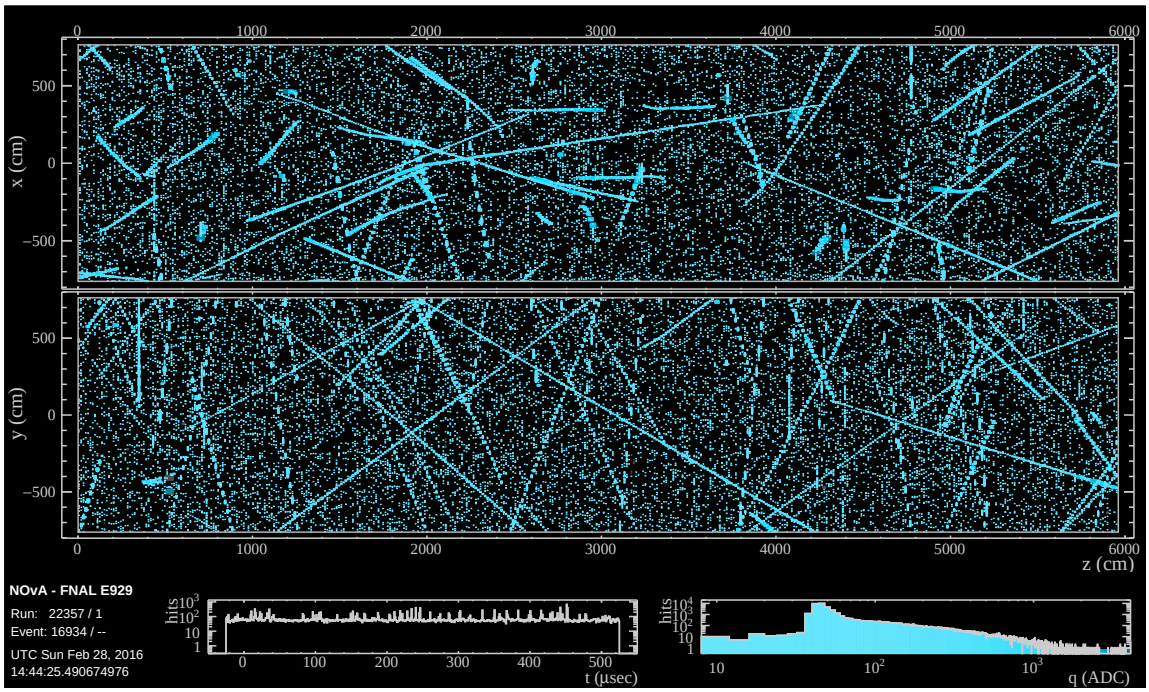


Figure 3.9: Example of the Far Detector NOvA Event Display. The hits are colored by the amount of charge deposited.

large sample of neutrino interactions. With a large sample of neutrino interactions, NOvA will be sensitive to the smallest variations in the neutrino oscillation probability (2.13). In particular, the NOvA experiment will be sensitive to the CP violation phase δ_{CP} and the sign of the neutrino mass difference Δm^2_{32}.

Since the NuMI beam will run half of the time in the FHC mode and another half of the time in the RHC mode, the NOvA experiment will be able to observe two neutrino oscillation modes $\nu_\mu \to \nu_e$ and $\bar{\nu}_\mu \to \bar{\nu}_e$. As was shown in Equation 2.14, the difference between $P(\nu_\mu \to \nu_e)$ and $P(\bar{\nu}_\mu \to \bar{\nu}_e)$ depends directly on the δ_{CP}. Therefore, by estimating these probabilities NOvA can make inferences about the value of the δ_{CP}.

Additionally, since neutrinos pass through Earth as they travel from the Near to Far Detectors the matter effect will come into play. As was shown in subsection 2.1.6, the presence of the matter effect renormalizes neutrino oscillation parameters. The magnitude of the renormalized neutrino oscillation parameters depends on the sign of Δm^2_{32}. Therefore, by measuring neutrino oscillation probabilities $P(\nu_\mu \to \nu_e)$ and

$P(\bar{\nu}_\mu \to \bar{\nu}_e)$ NOvA may be able to resolve the sign of Δm^2_{32}.

However, under certain conditions, it may happen that the matter effect will cancel the effect produced by the $\delta_{\rm CP}$, or they may amplify each other. Whether the matter effect will amplify or cancel the observed $\delta_{\rm CP}$ effect will depend on the values of $\delta_{\rm CP}$, Δm^2_{32} and θ_{23}. To illustrate complex interactions between these parameters one can make a bi-probability plot like Figure 3.10 that shows possible values of the estimated probabilities $P(\nu_\mu \to \nu_e)$ and $P(\bar{\nu}_\mu \to \bar{\nu}_e)$ depending on various neutrino oscillation parameters.

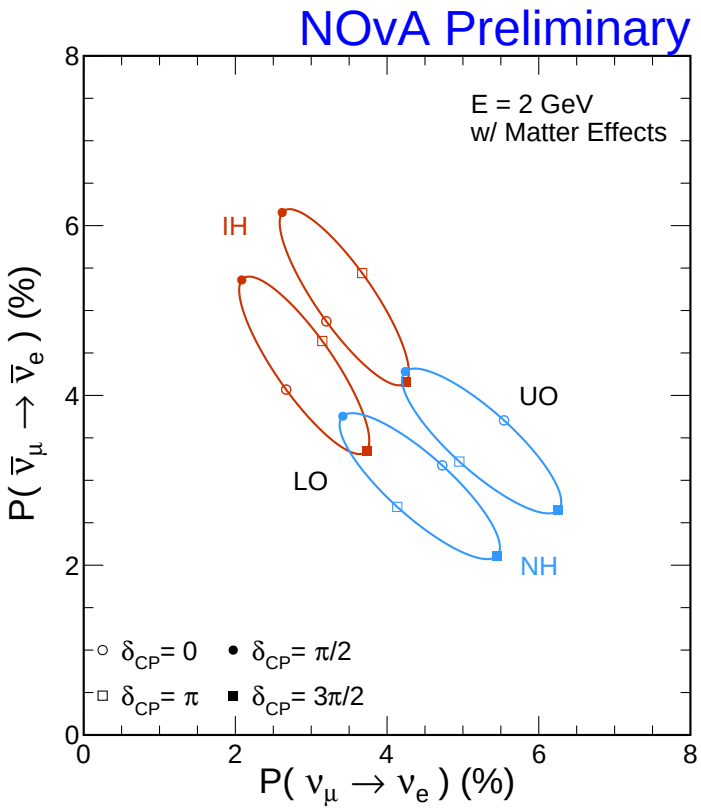


Figure 3.10: Example of a bi-probability plot for the NOvA experiment. The horizontal and vertical axes show the probability of neutrino and antineutrino oscillations. The contours represent possible values of measurements of these oscillation probabilities, depending on the octant of θ_{23} (UO – upper octant, LO – lower octant), sign of Δm^2_{32} (NH – normal hierarchy $m_3 > m_2$, IH – inverted hierarchy $m_3 < m_2$), and the CP violation phase $\delta_{\rm CP}$. The $\delta_{\rm CP}$ changes continuously in each contour from 0 to 2π.

The bi-probability plot Figure 3.10 shows that if the true neutrino oscillation parameters put neutrino oscillation probabilities somewhere in the intersection of the contours in Figure 3.10 (e.g. intersection of red and blue contours) then the NOvA experiment will not be able to tell which contour a given value of neutrino oscillation probabilities belongs to. On the other hand, if the true value of the neutrino oscillation parameters puts neutrino oscillation probabilities on the outer part of these contours, then the NOvA experiment will be able to resolve the contour and find the θ_{23} octant, the sign of Δm^2_{32}, and make an estimate of the δ_{CP}.

Chapter 4

Detector Calibration and Event Simulation

The primary goal of the NOvA experiment is to refine our understanding of the neutrino oscillation parameters. In order to get an estimate of the neutrino oscillation parameters NOvA uses the following strategy: (1) we measure neutrino event counts at the Near Detector; (2) use neutrino oscillation model (2.13) to get predicted oscillated neutrino counts at the Far Detector from the counts at the ND; (3) find values of the neutrino oscillation parameters that give the best agreement between the predicted FD neutrino counts and the measured FD event counts.

The high-level description above hides several important steps. First, the detector electronics can provide only very low-level information about a measured number of photo-electrons for each detector cell vs time. Starting with this low-level information we need to identify separate neutrino interactions to be able to count them.

Second, neutrino oscillation probabilities (2.13) depend on the energy of neutrinos. Correspondingly, in order to use the neutrino oscillation model to predict the FD neutrino event counts from the ND counts we need to estimate the energies of the neutrinos. The energy estimation of neutrinos is not a straightforward process, since there is no simple relation between the number of photo-electrons measured by APDs and the energies of the particles that produced these photo-electrons.

In order to simplify the process of energy estimation of neutrinos, NOvA performs a Calibration procedure. The goal of this procedure is to determine the energy

deposited by a particle in a given NOvA cell from the measured number of photo-electrons. This procedure also calibrates away any differences in photo-electron response between various detector cells. After the Calibration procedure, we will have a clear picture of the amount of energy deposited in each detector part vs time.

Next, we can use the calibrated information about the energy deposition vs time to identify separate neutrino interaction events and estimate their energies. To assist with the development of algorithms capable of detecting neutrino interactions and estimating their energies, NOvA uses samples of simulated neutrino interactions. These samples are generated for both Near and Far detectors using Monte Carlo techniques. The simulated samples have an advantage over the real data since we know exactly what particle deposited a given amount of energy, and the true kinematic properties of all particles.

I will begin this chapter by describing the Calibration procedure for the NOvA detectors. Then, I will discuss the details of the Monte Carlo simulation.

4.1 Calibration

There are two major components of the detector calibration: energy calibration and timing calibration [16]. The purpose of the energy calibration is to find a way to convert the raw cell readouts (ADC) from the APD into the energy that was deposited in the cell by particles. The purpose of the timing calibration is to calibrate out tiny clock differences between different NOvA FEBs and get the whole picture of the detector with the correct timing information. I will start the discussion of the calibration procedures with the energy calibration.

4.1.1 Energy Calibration

The NOvA detectors are made of multiple long cells (Figure 3.5) that capture light deposited in the scintillator with a wave-length shifting fiber. However, as light travels through the fiber it is attenuated. Therefore, the scintillator light captured near the top of the cell (Figure 3.5) will experience smaller attenuation and produce higher photodiode repose, compared to the light captured near the bottom of the cell

(due to having to travel a smaller distance in the fiber). The attenuation calibration aims to calibrate away this difference and convert the raw APD readouts into an intermediate quantity – the corrected number of photo-electrons (PECorr), which will be independent of whether the energy was deposited at the top of the NOvA cell or the bottom.

The attenuation calibration also ensures that the corrected number of photo-electrons is consistent across all detector cells. That is, it ensures that a given amount of energy produces the same amount of PECorr irrespective of the detector cell.

Unfortunately, as the detector ages, the properties of the cell response vary with time. This variation can make the previous attenuation calibration no longer valid. To address the detector aging a separate calibration is performed – the drift calibration.

Finally, once we get the corrected number of photo-electrons (PECorr) that is proportional to the energy deposited in the detector cells, we need to convert this value into the actual amount of energy deposited. The absolute energy calibration aims to achieve this task.

Cosmic muons are used in order to perform various energy calibrations since they have a very well understood energy deposition rate in the detector material.

Attenuation Calibration

The attenuation calibration starts with raw readouts (ADC) of the APDs. Equal amounts of energy deposited at different locations in a single detector cell will result in different ADC readouts due to light attenuation in the optical fiber. The attenuation calibration is performed on a cell by cell basis and it tries to calibrate out these differences and convert the ADC readout into a corrected number of photo-electrons (PECorr) that is independent of a location where energy was deposited.

The core assumption of the attenuation calibration is that muons deposit energy at an approximately equal rate as they travel through the detector. If this assumption is correct, then we can find a distribution of ADC per length that muon traveled in a cell (ADC/cm) vs location in the cell (W). This distribution can be used to normalize the ADC response and make it independent of the position in the cell

(W).

In order to construct a distribution of (ADC/cm) vs (W), we need to know the length that a given muon traveled in a detector cell and the position at which this muon crossed the cell. To estimate these quantities we need to select a sample of muons with faithfully reconstructed 3D tracks. Once such a sample is selected and the distribution of (ADC/cm) vs (W) is found, we fit a profile of that distribution with Equation 4.1.

$$y = C + A\left[\exp\left(\frac{W}{X}\right) + \exp\left(-\frac{L+W}{X}\right)\right] \tag{4.1}$$

where L is the length of the detector cell, and C, A, X are the fit parameters.

The fits of the form (4.1) are performed for each detector cell independently. However, they correctly describe the behavior of the (ADC/cm) vs (W) only away from the cell edges. At the edges of the cells, the behavior of the ADC response does not follow any recognizable pattern which is likely a result of a fiber twisting. In order to build a profile of ADC response near the edges of the cell, a correction is added to the profile (4.1). Figure 4.1 shows an example of such fit with a correction to a single Near Detector cell.

Usage of profiles of (ADC/cm) vs (W) obtained for each detector cell allows us to normalize the response of (ADC/cm) that is specific to each cell and convert it to a quantity (PECorr/cm) that is uniform across the entire detector.

Absolute Energy Calibration

Using normalized (ADC/cm) response expressed in terms of (PECorr/cm) we can work with the detector as a whole, without worrying about different responses of different cells. The next step of energy calibration is to convert this normalized PECorr into the actual energy that was deposited in the detector.

In order to convert the (PECorr/cm) into the deposited energy NOvA uses cosmic muons with well reconstructed tracks. The reason for using muons is that their energy deposition rate dE/dx is well studied and described by the Bethe-Bloch equation [17]. This means, that for muons dE/dx predicted by the Bethe-Bloch equation should be proportional to (PECorr/cm). And the coefficient of proportionality will give the

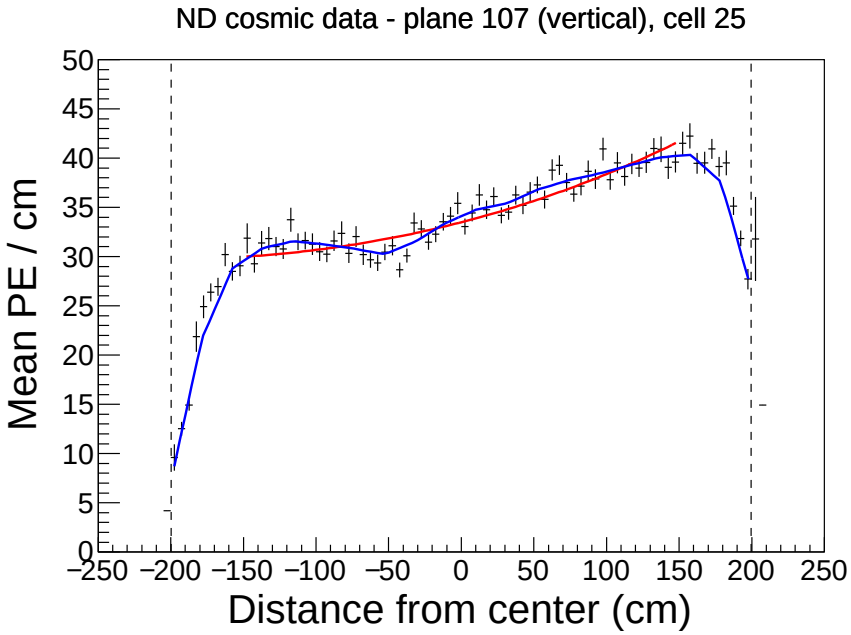


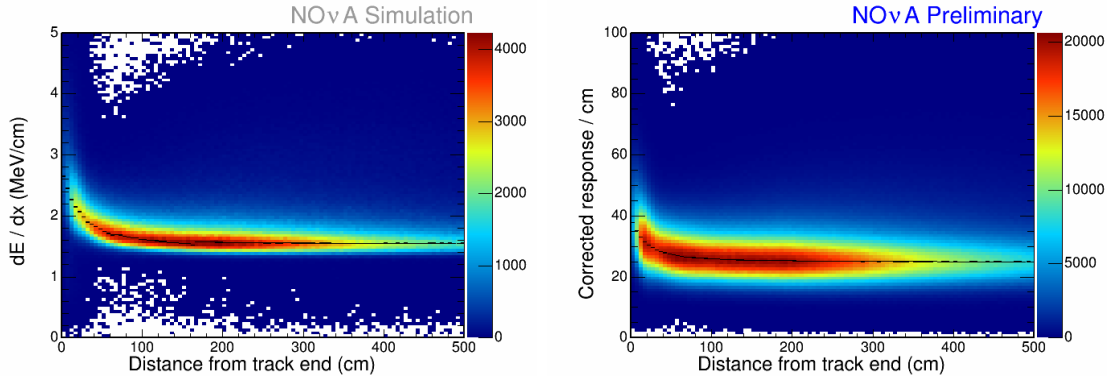
Figure 4.1: Example of an attenuation profile for a single vertical cell at the Near Detector. The red curve shows the double exponential fit (4.1). The blue curve shows the double exponential fit with the edge correction.

desired conversion factor from (PECorr) into the energy.

The Bethe-Bloch equation says that the muon energy deposition rate dE/dx is determined by the energy of the muon and by the properties of the material in which it travels. Therefore, to predict dE/dx by the Bethe-Bloch formula we need to know the energy of the muon first.

In order to find the energy of the muon NOvA considers muons that stop in the detector. If muon has stopped then we know that its kinetic energy is zero. Now we can start from the muon's stopping point and backtrack along its trajectory, integrating dE/dx predicted by the Bethe-Bloch equation. In this way, muon's energy can be estimated for all points along its trajectory.

Having found the energy of the muon along its trajectory, we can use the Bethe-Bloch equation to predict muon's energy loss dE/dx and use this dE/dx to find a conversion factor from the detector response PECorr/cm to the dE/dx. Figure 4.2 shows a comparison of the simulated dE/dx to the measured PECorr/cm for the cosmic muons.



(a) Simulated dE/dx (b) Measured PECorr/cm

Figure 4.2: Comparison of the simulated muon energy deposition calculated using the Bethe-Bloch equation (left) to the measured muon energy deposition expressed in terms of the detector response PECorr/cm (right).

Drift Calibration

The absolute energy calibration was performed under the assumption that the normalized detector response (PECorr/cm) stays the same over time. This assumption is not correct since as the detector is getting old, the scintillator is being slowly degraded, materials of the detector are aging and the response of the detector electronics changes. Therefore, another type of calibration is performed in order to correct for the drift of the detector response.

The drift calibration is performed for each detector channel separately. Similar to the other energy calibrations it also relies on cosmic muons. The drift calibration tracks the mean of PECorr/cm deposited by cosmic muons for each cell. Changes in the mean of PECorr/cm with time indicate a change of the detector response with time. This change is monitored and adjusted by the drift calibration.

4.1.2 Timing Calibration

The NOvA Far Detector receives a beam of muon neutrinos in pulses of 10 µs each 1.3 s. Due to the beam divergence and the off-axis alignment we can expect only about $1-2$ neutrino interactions in the Far Detector per day. On the other hand, the Far Detector has a high rate of cosmic muons (about 120 000 Hz). With such

a large discrepancy between the signal and background rates, no data driven neutrino trigger is possible, and therefore we rely on timing information provided by Fermilab (the NuMI trigger), in order to select timing windows when the beam is on. For the timing information from Fermilab to be useful, both detectors must be carefully synchronized with the global time, which is one of the goals of the Timing Calibration.

Another goal of the Timing Calibration is to estimate timing resolutions of the detector hits since multiple reconstruction algorithms at NOvA rely on this timing resolution information for proper operations.

Online Calibration

When NOvA detectors are active, their clocks are continuously synchronized with the GPS time. This synchronization process is also known as Online Timing Calibration at NOvA.

The process of the online timing synchronization is closely tied to the architecture of the Data Acquisition System (DAQ) described in section 3.3. For context, APD arrays are mounted on FEBs. Each FEB services 32 APDs and has an independent clock. The signal from 64 FEBs is collected by the Data Concentrator Module (DCM) and then relayed to the buffer node farm. The DCMs are synchronized to the master clock in each detector. It is the responsibility of each DCM to further synchronize its clock with the FEBs that it services.

In order to synchronize all FEBs in one DCM, the DCM continuously distributes its clock among the FEBs that it services. To make sure that all FEBs receive the clock value at the same time, a requirement on lengths of cables connecting DCMs to the FEBs is placed, namely that all cables should have the same length. This ensures, that all FEBs in one DCM receive the clock information at the same time. Moreover, signal transmission delays in cables are measured and accounted for to ensure correct clock synchronization.

The clocks belonging to different DCMs are synchronized to the master clock of the timing system. The timing system continuously pings DCMs and measures their response time to estimate and correct the delays caused by signal propagation in cables from the DCMs to the master clock. The master clock itself has GPS

receivers and is synchronized to the GPS time.

Offline Calibration

As a cross check for the online clock synchronization, the offline timing calibration is performed as well. For this purpose, a sample of well reconstructed muons traveling through the detector is selected. If clocks are synchronized correctly then all hits caused by a single muon should occur at the same time (corrected for the muon time of flight and signal propagation in the fiber). By measuring the difference in muon hit times between different DCMs over a large sample we can verify that there is no timing miscalibration between the DCMs. In case if such miscalibration is detected between DCMs we can correct it using the measured difference in hit times.

Timing Resolution

The estimation of timing resolution for each DCM is performed in the offline mode. For this purpose, the same sample of muons is used as for the offline timing calibration. The 2D histogram of differences in hit times between two different channels in a single DCM (similarly corrected for the muon time of flight and signal propagation in cables) vs the recorded value of ADC is constructed. Figure 4.3 shows an example of such a histogram. The timing resolution is then determined from the histogram, by taking an RMS value of hit times for each value of ADC.

4.2 Simulation

The primary goal of the NOvA simulation is to generate samples of simulated neutrino interactions in the NOvA Near and Far detectors. The generated samples should have enough statistics in order to:

- Perform comparisons between Data and Simulated samples at the Near Detector to verify the validity of the physical model used in the simulation.
- Train and test reconstruction algorithms for the Near and Far detectors.

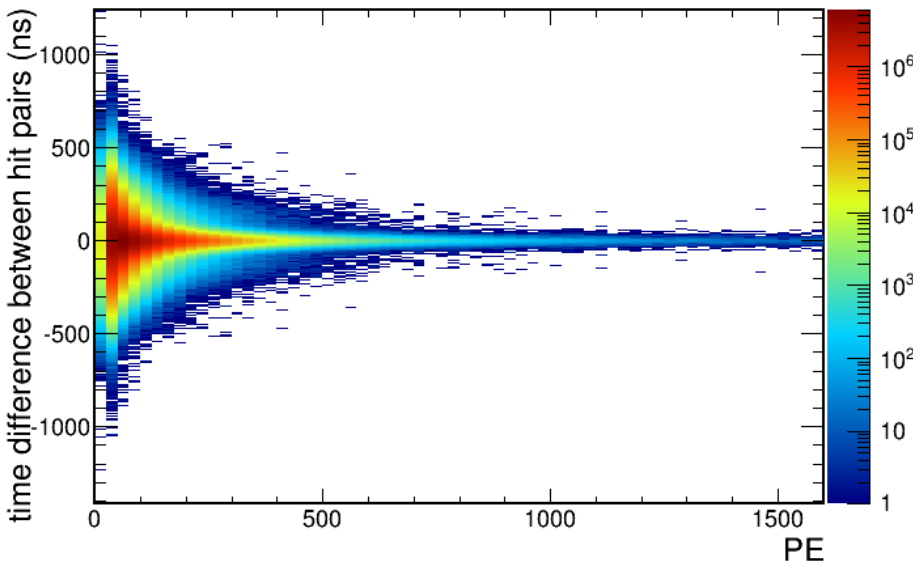


Figure 4.3: Example of the Far Detector timing resolution histogram for a single DCM.

In this section, I will describe how the generation of the simulated neutrino interaction samples (also called Monte Carlo samples) is performed. The entire procedure can be roughly summarized as a series of 4 steps, where results of the previous step are used to initialize the following step. The four steps are:

1. Generation of neutrinos in the NuMI decay pipe. At the end of this step, we receive kinematic information of neutrinos and their flavors.

2. Simulation of neutrino interactions with the detector. At the end of the second step, we receive information about whether the neutrino interacted with the detector, and if so then we get a list of primary particles produced in such interaction.

3. Simulation of propagation of the primary particles through detector medium. As primary particles are propagated through the detector they deposit energy and may create cascades of secondary particles. The result of the third step is information about energy deposition by particles in the detector cells.

4. Simulation of the detector and electronics response to the deposited energy.

4.2.1 Beam Simulation

The first step of the simulation chain is the simulation of the NuMI beamline. It is performed by the FLUKA/FLUGG software developed by Fermilab [18]. The FLUGG software package simulates interactions of 120 GeV protons with the graphite target. Then, it propagates hadrons, produced in these interactions, through the magnetic horns and tracks them in the decay pipe. It records all hadron decays that produced neutrinos and saves the resulting neutrinos (called neutrino "rays"). Each saved neutrino "ray" contains information about the flavor of the neutrino, its energy, and momentum.

4.2.2 Simulation of Neutrino Interactions

The NOvA experiment relies on the GENIE software [19] to simulate neutrino interactions with the detector. GENIE is a Monte Carlo simulation package, which tries to probabilistically predict neutrino interactions with the surrounding medium based on measured interaction cross-sections.

The GENIE package receives as input information about neutrino flavor and its kinematic properties from the simulated beam "rays". In addition to that, it requires a detailed detector geometry, which was carefully determined by the NOvA group.

GENIE performs Monte Carlo sampling in order to predict whether the neutrino interacted with the detector. If a neutrino interaction with the detector is sampled, then GENIE will output an interaction vertex and a list of daughter particles. Furthermore, it will model interactions of the daughter particles with the nucleus at the interaction vertex.

As a result of the event simulations with GENIE, we will get a list of neutrino interactions with the detector, where each interaction will have a well defined vertex and a list of daughter particles with their kinematic properties.

4.2.3 Simulation of Propagation of Daughter Particles

To propagate daughter particles predicted by GENIE through the detector, NOvA uses GEANT4 software [20]. GEANT4 propagates the daughter particles step by step through the detector medium. Similar to GENIE it also relies on a precise detector geometry and uses Monte Carlo methods to simulate interactions of the daughter particles with the environment.

GEANT4 tracks all daughter particles until they lose all their energy or until they leave the detector volume. As the daughter particles travel through the detector and interact with its atoms they can create secondary particles or simply decay. GEANT4 is capable of simulating such processes as well. At the end of the propagation process, GEANT4 produces total energy deposits in the scintillator created by all particles.

4.2.4 Simulation of the Detector Response

The last step of the simulation takes energy deposits produced by GEANT4 and converts them into scintillator light. Then, it transports the scintillator light into the APD and simulates APD and electronics response to the light. Finally, the resulting outputs are converted to a form similar to that of the real data.

Deposited Energy to Scintillator Light Conversion

If the NOvA scintillator was perfectly linear then the light yield will be proportional to the deposited energy. However, it was discovered that at high energies organic scintillators saturate and the light yield is reduced compared to a simple linear model. To describe the light yield of the scintillator NOvA uses the Birks-Chou model [21]:

$$\frac{dL}{dx} = \frac{L_0 \frac{dE}{dx}}{1 + k_B \frac{dE}{dx} + k_C \left(\frac{dE}{dx}\right)^2} \tag{4.2}$$

where k_B, k_C are the Birk and Chou correction factors and L_0 is a proportionality coefficient between the deposited light $\frac{dL}{dx}$ and the deposited energy $\frac{dE}{dx}$ in the linear regime of the scintillator. The correction factors are determined experimentally by matching Near Detector Data to MC.

The scintillator light is not the only type of light that can be produced in the NOvA detectors. When charged particles travel through the medium they can create Cherenkov radiation. The intensity of the Cherenkov radiation is dependent on the particle speed and hence it cannot be modeled by the Birk-Chou model. In order to get a faithful light model, NOvA also simulates Cherenkov light production [22]. The simulation of the Cherenkov radiation improves Data to MC agreement and gives more reasonable values for the Birk and Chou coefficients.

Photon Transport in Scintillator

In order to propagate the light generated in the scintillator to the APDs, NOvA employs a ray tracing simulation. The ray tracing simulation tries to trace light rays from the point where they originate until they enter the wavelength shifting fiber. Using ray tracing we can obtain a two dimensional histogram like Figure 4.4 that describes the photon collection probability by a fiber as a function of a distance between the emission and collection points and a time difference between emission and collection.

Then, the histogram shown in Figure 4.4 is used to propagate light from a given point in the detector cell to the wavelength shifting fiber.

Photon Transport in Fiber

After photons have reached the wavelength shifting fiber, their propagation in fiber is simulated as well. It is assumed that on average half of the time photons will be traveling in the direction towards the APD and another half in the opposite direction (c.f. Figure 3.5). Correspondingly, when the light enters the fiber, half of the light is transported towards the APD and another half is transported in the opposite direction.

The number of photons surviving the transport over the fiber is determined using the attenuation curves (e.g. Figure 4.5). These curves tell us what fraction of photons reached the APD as a function of distance from it.

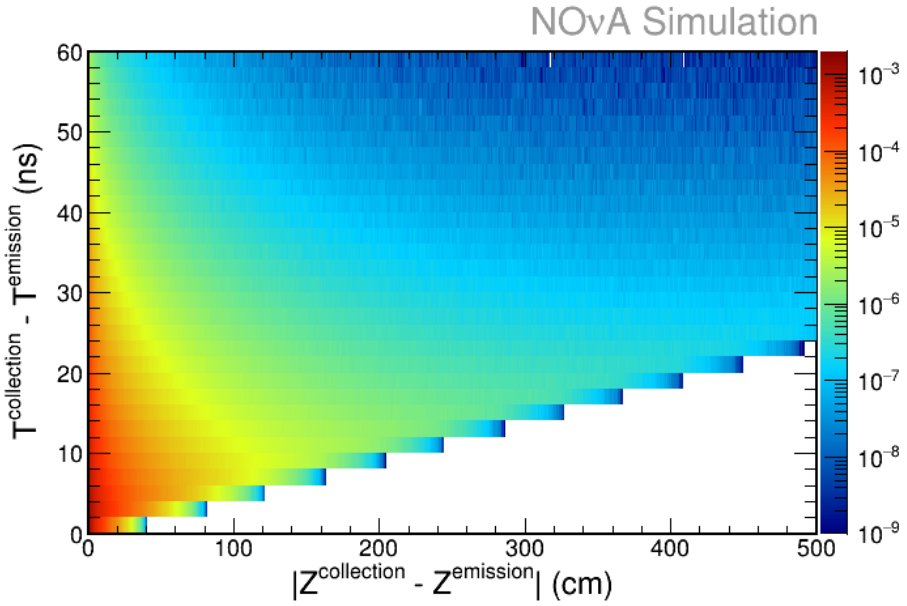


Figure 4.4: Two-dimensional histogram of the light tracing simulation. The horizontal axis shows the distance along the direction of the cell between a point where the light was emitted and a point where the light entered the wavelength shifting fiber. The vertical axis shows the difference between the emission time and the collection time.

Simulation of the Electronics Response

At the end of the previous steps, we obtain a number of photons reaching APDs as a function of time. Next, NOvA simulates the response of APDs and electronic boards. This simulation is a very complex and multistage process that goes well beyond the scope of my thesis. The full details of this process can be found in [18]. In this section, I will describe only the high level overview of the electronics response simulation.

First, a simulation of the conversion of the number of photons received by the APD into the number of induced photo-electrons is performed. The measured quantum efficiency of APDs is used for this purpose. In addition to that, an internal APD noise is simulated and added to the generated signal.

Next, a simulation of the conversion of the generated APD response into a digitized signal in the FEB is performed. This simulation includes simulation of trans-

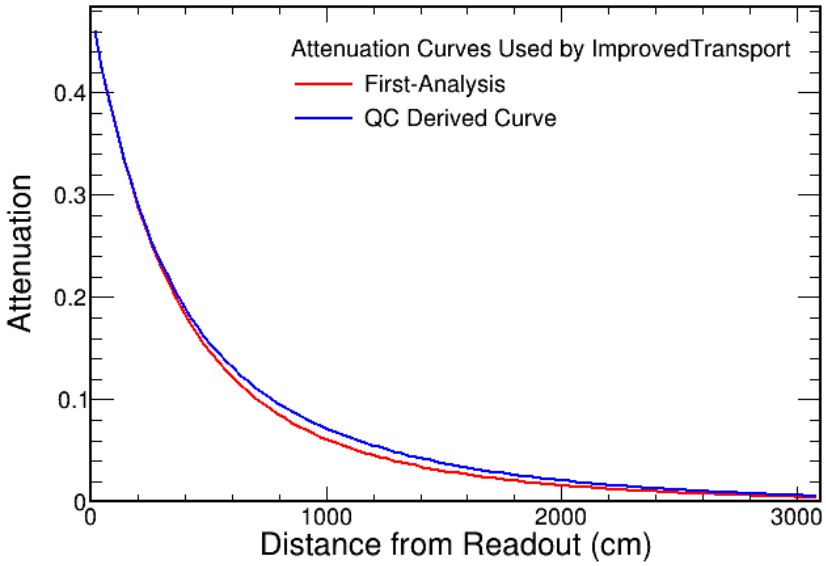


Figure 4.5: Example of a light attenuation curve used to propagate photons through the fiber. The red curve is the old curve that was used in the First NOvA Analysis. The blue curve is the revised curve.

mission of the APD signal through the analog parts of the FEBs, digitization of the analog signal by ADC, and correction of artifacts caused by electronic boards.

Finally, after the response of the electronics is simulated the results are converted to the same format as used for Data files and stored for further analysis.

Chapter 5

Event Reconstruction

The raw detector data is just a collection of hits – readings of photo-electron counts from the detector cells. Each hit contains information about the 2D spatial coordinates of the detector cell in which it occurred, the time of the reading, and the total energy deposited in the hit.

In order to proceed with the estimation of the neutrino oscillation parameters, we need to reconstruct a complete picture of a neutrino interaction in the detector and answer two main questions about the hits in the detector:

1. What kind of particle produced a given set of detector hits.

2. What was the energy of the particle that produced these hits.

Correspondingly, the ultimate goals of the reconstruction are to identify particles that created the detector hits and to estimate energies of these particles. The particle identification (PID) and energy estimation tasks turn out to be too difficult to be performed on the raw detector hits. Therefore, many algorithms are run to reconstruct higher-level objects (clusters of hits, particle tracks, etc) that are more tractable than the raw detector hits. These intermediate objects are then used to simplify the particle identification and energy estimation tasks.

In this chapter, I will describe the standard NOvA reconstruction chain. This chain starts with the raw detector hits and produces information about particles that interacted in the detector and their energies.

5.1 Hit Clustering

The standard NOvA reconstruction chain begins by grouping detector hits that are "close" in space-time into clusters. Then, it runs more sophisticated algorithms to detect "features" in the clusters of hits. Next, using these features it tries to identify sub-clusters of hits that belong to the individual particles in the detector.

5.1.1 Slicing

The first step of the clustering chain is called Slicing. The slicing procedure tries to identify clusters that contain hits of all particles produced in a single interaction. For the slicing procedure, a single interaction is defined as either an interaction of a neutrino with the detector or a cosmic ray.

The single NuMI trigger readout window lasts 500 µs. With the rate of cosmic muons at the Far Detector of 120 000 Hz, we can expect at least 60 particle interactions inside the detector in a single readout. The slicing algorithm needs to be able to separate these interactions into different clusters.

The slicing algorithm is based on DBSCAN (Density-based spatial clustering of applications with noise [23]). DBSCAN begins by calculating pairwise distances between each detector hit. If some hit has at least N_{min} neighbors at distances less than ϵ, then DBSCAN calls such a hit a core point.

The slicing algorithm proceeds by joining together core points that are within a distance of ϵ of each other, thereby forming clusters. It continues to join neighbor core points into clusters until no more neighbor points can be found.

Finally, once the core points have been partitioned into clusters, the slicing algorithm expands the clusters by adding boundary points (B) to them. That is, points (B) that are not core points, but that lie within the ϵ-neighborhood of some core point (C).

The resulting clusters of hits are called slices at NOvA. The slicing algorithm as described above has several tunable parameters. They are ϵ, N_{min} and a function that computes the distance between two hits. These parameters are fine-tuned in order to make sure that the slicing algorithm gives good separation between different interactions and does not aggressively divide a single interaction into multiple

clusters.

5.1.2 Identification of Linear Features

When a neutrino interacts with a nucleus of the detector we expect there to be a single interaction vertex from which all primary daughter particles emanate. In order to help to find this interaction vertex, another step in the reconstruction chain is performed. This step tries to identify major linear "features" of the event. The motivation for this step is that the intersection of these linear "features" should give us the interaction vertex.

In order to find the linear features in the event, the Hough Transform method [24] is used. The Hough Transform method starts by iterating over all pairs of hits in the event and building straight lines that pass through these hits. The straight lines are parametrized by (ρ, θ) using the Hesse normal form (5.1):

$$\rho = x \cos\theta + y \sin\theta \tag{5.1}$$

where (x, y) are the coordinates of the points along the line. Geometrically, ρ is the distance from the coordinate origin $(0, 0)$ to the closest point on the line. θ is the angle between the x-axis and a segment that connects the coordinate origin to the closest point on the line.

The Hough Transform method considers a set of all lines $\mathcal{L}$ built through pairs of hits. It constructs a 2D density map M in a space of line parameters (ρ, θ). In order to build this map, the Hough Transform iterates through the set $\mathcal{L}$, and for each line parametrized by (ρ_0, θ_0) in this set, it adds a Gaussian distribution (5.2) centered at (ρ_0, θ_0) to the density map M.

$$P(\rho, \theta) = \exp\left(-\frac{(\theta - \theta_0)^2}{2\sigma_\theta^2}\right) \exp\left(-\frac{(\rho - \rho_0)^2}{2\sigma_\rho^2}\right) \tag{5.2}$$

where $\sigma_\rho = 3\,\mathrm{cm}/\sqrt{12}$, $\sigma_\theta = 3\,\mathrm{cm}/d\sqrt{6}$, and d is the distance between points that were used to construct a line.

When the Gaussian map M is built it is further smoothed out by applying a local gaussian blur.

Next, a linear feature extraction phase begins. In order to identify linear features in the event, peaks of the density map M are found. The peak search algorithm simply finds the tallest peak in the density map M. Coordinates (ρ_0, θ_0) of that peak define a major linear feature of the event. Once the peak is found, all hits that are related to the peak are removed from the consideration and the density map M is rebuilt without those hits.

The search of linear features is repeated until there are no peaks left that are higher than some predefined threshold, or until the predefined maximum number of linear features has been found. This linear feature identification procedure is run separately for two detector views – XZ and YZ.

5.1.3 Identification of the Interaction Vertex

In order to find an interaction vertex NOvA uses the Elastic Arms method [25] (also known as a method of deformable templates [26]). The Elastic Arms method identifies the interaction vertex and major event "arms" (3D straight lines emanating from the interaction vertex). That is, the Elastic Arms algorithm tries to find a set of lines defined by Equation 5.3:

$$\begin{aligned} x_\alpha(s) &= x_0 + s \sin\theta_\alpha \cos\phi_\alpha \\ y_\alpha(s) &= y_0 + s \sin\theta_\alpha \sin\phi_\alpha \\ z_\alpha(s) &= z_0 + s \cos\theta_\alpha \end{aligned} \tag{5.3}$$

where (x_0, y_0, z_0) are the coordinates of the interaction vertex, $(\theta_\alpha, \phi_\alpha)$ are the angles that specify the direction of the arm α and s is the parameter of the line. In order to find such a set of lines, the Elastic Arms method tries to minimize the following cost function (5.4):

$$E = \sum_i \sum_\alpha V_{i\alpha} M_{i\alpha} + \lambda \sum_i \left(\sum_\alpha V_{i\alpha} - 1 \right)^2 + \lambda_V \sum_\alpha D_\alpha \tag{5.4}$$

where $M_{i\alpha}$ is the distance between hit i and arm α; $V_{i\alpha}$ is the strength of association between hit i and arm α; D_α is the distance from the vertex to the first hit in arm

α; λ, λ_V are the hyperparameters of the algorithm. The sum over i is the sum over all hits in the event, and the summation over α is the summation over all arms.

Minimization of the first term in (5.4) minimizes distances between hits and arms to which they are associated. The second term adds a penalty to hits that have not been associated with any arm. The final term ensures, that for events having a single particle (and a single elastic arm) an interaction vertex will be placed at the first hit on the arm (otherwise it can be placed anywhere on the arm creating an ambiguity).

The association $V_{i\alpha}$ between hit i and arm α takes into account the fact that some hits can be due to noise and therefore need not be associated with any arm. It is given by Equation 5.5, where β is a parameter of the Elastic Arms algorithm. The term $\exp(-\beta\lambda)$ was introduced in the denominator to lower the association of the noise hits to the elastic arm.

$$V_{i\alpha} = \frac{\exp(-\beta M_{i\alpha})}{\exp(-\beta\lambda) + \sum_{\alpha'} \exp(-\beta M_{i\alpha'})} \tag{5.5}$$

The cost function (5.4) is optimized by a method of simulated annealing [27]. Parameter β, introduced in the definition of $V_{i\alpha}$, plays the role of the inverse of the annealing temperature $T = 1/\beta$. In order to seed the initial locations of the elastic arms, linear features found in the previous section are used [25]. The number of elastic arms to seed is determined by taking the maximum number of linear features found in either XZ or YZ detector views.

The simulated annealing procedure is restarted with various vertex seeds, and the best location of the elastic arms is selected among them, according to the cost function (5.4).

After the best location of the elastic arms is identified, coordinates of the interaction vertex are given by the parameters (x_0, y_0, z_0) (c.f. Equation 5.3).

5.1.4 Finding Clusters of Hits Describing Individual Particles

The next step in the reconstruction chain is to form clusters of hits in such a way, that each cluster is associated with a single particle in the event. To achieve this objective

NOvA relies on the fact that each event has a distinctive interaction vertex found in the previous step. All particles in the event effectively emanate from this interaction vertex. Therefore, when placing an observer at the very interaction vertex, hits produced by a single particle would be closely located in a field of view.

An algorithm called Fuzzy-K [28] was designed for the purpose of finding such clusters of hits. To perform the clustering, this algorithm starts with finding angles of hits (and their uncertainties) when looking from the interaction vertex. This procedure is performed separately in each detector view XZ and YZ. Then, the angles of hits are clustered in each detector view using a possibilistic Fuzzy-k-Means clustering algorithm [29].

The possibilistic Fuzzy-k-Means algorithm is a modification of the original Fuzzy-k-Means algorithm [30] that relaxes a requirement that a given hit should belong to some cluster with a probability equal to 1. Relaxing this requirement allows noise hits to not be associated with any clusters.

The possibilistic Fuzzy-k-Means algorithm requires knowledge of the number of clusters (k) in advance. Since we do not have such knowledge, the Fuzzy-k-Means algorithm is run iteratively multiple times, starting with a single ($k = 1$) global cluster and adding a new cluster each time. This process stops if all hits have been associated with clusters or when the maximum number of clusters is reached.

Once the 2D clusters have been identified by the Fuzzy-k-Means in each detector view (XZ, YZ), another algorithm [28] is run to match pairs of 2D clusters into 3D clusters. In order to perform such matching, profiles of cumulative energy vs distance from the 2D cluster origin are built in each detector view. Then, the profiles of cumulative energy vs distance are matched pairwise between each detector view, and the best 2D matching clusters are merged into a single 3D cluster.

The clusters of hits that have been successfully matched between the detector views are called prongs (or 3D prongs) at the NOvA experiment. The 2D clusters that were not matched are called 2D prongs.

5.1.5 Clustering Summary

In this section, I have described the three main clustering steps forming the basis of the NOvA reconstruction chain. These clustering steps achieve the following

objectives:

- All activity in the detector is separated into Slices that contain only hits belonging to a single event (neutrino interaction or a cosmic ray).
- An interaction vertex is found in each slice.
- Each slice is then subdivided to find clusters of hits describing individual particles in each event.

5.2 Tracking Particles

Hit clustering algorithms allow NOvA to identify clusters of hits belonging to individual particles in each event. These clusters have a source point, given by the interaction vertex. From the source point, one can determine the direction of the cluster. But, apart from having a source point and direction, these clusters have no other internal structure. For many applications, however, it is useful to know the precise trajectory of the particle that produced a given cluster of hits.

Therefore, NOvA runs two tracking algorithms to identify tracks of particles within the clusters of hits:

1. A Kalman Filter algorithm is specifically designed to find tracks of μ in each slice.
2. A Break Point Fitter algorithm that is capable of finding tracks of $\pi^{\pm}$, μ, and p in each prong.

5.2.1 Muon Tracking with Kalman Filter

The muon tracking algorithm [31] consists of three steps. During the first step, hits that could form muon tracks are identified in each detector view. Next, the algorithm tries to fit muon tracks through the hits identified in the first step. These tracks are found independently for each detector view. Finally, fitted tracks from each detector view are matched to each other.

The first two steps rely on a Kalman Filter algorithm [32]. In general, a Kalman Filter algorithm is able to predict a true state of a model from a series of sequential observations (with uncertainties). The Kalman Filter algorithm is capable of such inference, provided that we know how the true state evolves from one observation to another.

In our case, the true state that we are trying to find is the true trajectory of a muon. Using a theoretical model we can predict how a muon should move from one point to the other, and it allows us to apply the Kalman Filter algorithm.

The muon tracking algorithm starts with identifying possible tracks in a slice. First, it forms track seeds by taking two hits that have a distance less than N_{seed} planes from each other. Then, the Kalman Filter algorithm is run on these seeds assuming that the particle travels in the upstream beam direction. The seeds are grown by adding new points if they are close to the predicted track. This closeness is determined by checking if a hit could be produced by a predicted track with a probability greater than P_{close}. If it can, then the hit is added to the seed and the Kalman Filter is run again to refine the predicted particle track. This process is repeated until there are no more hits that can be produced by any track with a probability greater than P_{close}.

After the upstream run of the Kalman Filter algorithm finishes, it is run again from the endpoints of the formed tracks in the downstream direction. This run is performed to pick up possibly missed hits. When possible tracks are identified the Kalman Fitter algorithm is run the third time to construct the final track predictions from the identified candidate tracks.

Finally, the tracking algorithm tries to match the fitted 2D tracks in each detector view to obtain the 3D tracks. This is achieved by comparing the difference of starting and ending z coordinates of the 2D tracks to the number of planes that the 2D tracks have in common. If the difference in start/end positions between a pair of 2D tracks is less than $S_{\text{threshold}}$ times the number of overlapping planes then these tracks are merged into a single 3D track.

5.2.2 Particle Tracking with Break Point Fitter

The Kalman Fitter tracking method described in the previous section is designed for the single purpose of finding muon tracks. To fit tracks of other particles NOvA uses another algorithm – Break Point Fitter [33]. The Break Point Fitter (BPF) algorithm assumes that as a particle travels through the detector it undergoes a finite number (M) of scattering events, each of which changes the direction of particle propagation. Given a collection of hits that were produced by a particle, BPF tries to find the location of points where the particle underwent scattering events and the corresponding changes of the particle trajectory. In other words, the BPF algorithm assumes that the particle trajectory can be expressed as Equation 5.6 and tries to optimize its parameters.

$$x(z) = \alpha + \beta z + \sum_{J=1}^{M} \alpha_J (z - Z_J)\, \theta\,(z - Z_J) \tag{5.6}$$

where $x(z)$ is the coordinate of particle trajectory orthogonal to the z axis; α, β – are the slope and intercept of the particle trajectory, α_J – are the scattering angles, Z_J – are the locations where the scattering occurred, and $\theta(x)$ is the Heaviside Step function.

The BPF algorithm is run for each 3D prong produced by the Fuzzy-k-Means algorithm (c.f. subsection 5.1.4). To find optimal values of parameters $\alpha, \beta, \alpha_J, Z_J$ the following objective function is minimized:

$$\chi^2 = \sum_i \frac{(x(z_i) - x_i)^2}{\sigma_i^2} + \sum_J \frac{\alpha_J^2}{\sigma_J^2} \tag{5.7}$$

where σ_i is the estimate of the error of measurement of a hit location, and σ_J^2 is the RMS value of α_J calculated from the theoretical expectation for the value of the scattering angle α_J. If positions Z_J at which scattering occurred are known, then the optimization of Equation 5.7 for α, β, α_J is trivial, since $x(z)$ is a linear function of these parameters. The nontrivial part of the BPF algorithm is finding the locations Z_J where the scattering occurred.

In order to estimate the locations where the scattering occurred, the BPF algo-

rithm finds the last point of a given 3D prong. Then, it goes from the end of the track looking at each hit. While backtracking, the BPF algorithm tries to estimate the energy of the particle by integrating the known shape of energy losses dE/dx in the detector materials. For such energy estimation to be possible, the energy of the particle needs to be known at the last point of the trajectory.

The energy of the particle at the last point of the trajectory has two components – kinetic energy and the rest energy of the particle. The BPF algorithm assumes that the particle has lost all its kinetic energy at the endpoint. Since there is no way to know the rest energy of the particle in advance, the BPF algorithm is simply run three times, assuming that the particle is either μ or $\pi^{\pm}$ or p.

Once the energy of the particle is estimated at each position in a prong the BPF algorithm tries to determine the scattering points Z_J. It places a new scattering point Z_J if either of two conditions occurs:

- If the distance from the last scattering point is greater than 1 m.
- If the deviation from a straight line predicted by the multiple scattering hypothesis is greater than 3 mm.

The first criterion here ensures that the scattering points are not too far apart from each other. The second criterion guarantees that the extrapolation of the track from the scattering point location is accurate up to a 3 mm which is a minimum resolvable detector distance (width of the cell wall).

After the scattering points have been placed, the objective function (5.7) is minimized for α, β, α_J. As the result, the BPF algorithm fits three tracks (under the assumption that the particle is either μ or $\pi^{\pm}$ or p) through each 3D prong in a slice. It also estimates particle energy for each track by summing the energy losses dE/dx in material.

5.3 Particle Identification

For some analysis tasks, it is useful to know the type of particle that created a given cluster of hits (prong). NOvA uses three algorithms that are capable of predicting the type of particle that produced a given prong. Two of these algorithms rely on a

statistical analysis of the rate of energy deposition dE/dx of a particle. The third algorithm uses images of prongs and predicts the particle type with the help of a Convolutional Neural Network.

5.3.1 Muon Identification with RemID

The RemID (Reconstructed Muon Identification) algorithm [34] is specifically designed to predict whether a given track was created by a muon. This algorithm is based on an observation that muons travel through the detector by almost straight lines and their rate of energy deposition is relatively small and approximately constant. Therefore, if one finds a long and straight track in the detector, then this track was likely left by a muon.

The only other particles that are capable of producing long tracks with similar energy deposition rates are the charged pions $\pi^{\pm}$. Charged pions are produced in large quantities in Neutral Current interactions. Most of the charged pions interact strongly with the detector nuclei. However, some of them manage to avoid interacting strongly with nuclei and travel through the detector interacting only electromagnetically. Since the mass of a charged pion ($\approx 140\,\text{MeV}$) is close to the mass of a muon ($\approx 106\,\text{MeV}$) they exhibit similar behavior under the EM interactions and leave similar tracks in the detectors.

The RemID algorithm aims to predict whether a given Kalman track (c.f. subsection 5.2.1) was produced by a muon or by a charged pion. RemID relies on four input variables to distinguish muon tracks from pion tracks: dE/dx Log-Likelihood (LL), $d\theta/dx$ (scattering angle) LL, track length, and a fraction of planes in a particle track that have contamination from hadronic activity.

The particle Log-Likelihoods dE/dx and $d\theta/dx$ are calculated as a sum of differences of Log-Likelihoods of the form:

$$LL^{dE/dx} = \sum_i \left(LL_{\mu}^{dE/dx}[i] - LL_{\pi}^{dE/dx}[i] \right) \tag{5.8}$$

where the sum over i is performed over the plane index, counting from the end of the track, $LL_{\mu}^{dE/dx}[i]$ is the LL that the dE/dx deposition at plane i was produced by a

muon, and $LL_{\pi}^{dE/dx}[i]$ is the LL that the dE/dx deposition at plane i was produced by a charged pion. The Log-Likelihoods at plane i are calculated by comparing the actual dE/dx to a probability distribution of dE/dx for a given particle type, which was constructed from a large sample of simulated particles.

The RemID algorithm performs particle classification with the help of the kNN method [35], which essentially compares how "close" input variables for a given particle to the corresponding variables of previously seen muons and pions. Based on the calculated closeness, RemID predicts a score for a given track to be produced by a muon vs charged pion.

5.3.2 Electron Shower Identification with ShowerLID

Compared to muons, electrons have a very small mass. Therefore, when e travels through the detector, it quickly loses its energy due to hard scattering off the atomic electrons in the detector volume. A high-energy electron traveling through the detector is going to spawn a shower of secondary electrons (from ionized atoms) and photons. Such showers are typically much shorter and wider than the tracks left by muons. To identify showers spawned by various particles NOvA uses another algorithm – ShowerLID [36].

The ShowerLID algorithm tries to predict whether a given shower was produced by an electron e or a photon γ or a muon μ or a neutral pion π^0 or a proton p or a neutron n or a charged pion $\pi^{\pm}$. At the start of the algorithm, it renormalizes hit energies for each 3D prong. This normalization is performed since the Fuzzy-k-Means algorithm can assign a given hit to multiple 3D clusters. Therefore, a simple summation of energies from each 3D prong can lead to double counting of hits. The normalization ensures that the energy from a hit is properly divided between the clusters to which it belongs.

To make a prediction about a particle type the ShowerLID algorithm relies on Log-Likelihoods of energy deposition, similar to the RemID algorithm. To calculate the Log-Likelihoods ShowerLID uses an equation like Equation 5.8 (for the RemID). There are two differences however from the RemID case:

- Since showers are wide, the energy deposition dE/dx is calculated separately

in two dimensions – along with the shower and transverse to the shower. ShowerLID uses both transverse and longitudinal Log-Likelihoods when predicting the particle type.

- RemID is only concerned about distinguishing muons from charged pions, therefore the only Log-Likelihoods that it uses are the differences between the muon LLs and charged pion LLs (5.8). However, ShowerLID is a multi-class classifier, therefore it looks at the differences between the Log-Likelihoods of electron vs Log-Likelihoods of $\gamma, \mu, \pi^0, p, n, \pi^{\pm}$.

For each shower, the ShowerLID algorithm produces the Log-Likelihoods for each particle type as its output. These Log-Likelihoods can be used to judge what kind of particle created a given shower.

5.3.3 Multiclass Particle Identification with Prong CVN

As an alternative to the Log-Likelihood approach, NOvA has another particle classifier based on a convolutional neural network (CNN). The Prong CVN [37] (convolutional visual network) algorithm was developed to take images of prongs (in XZ and YZ views) and predict a probability that a given prong was produced by an electron vs photon vs muon vs pion vs proton vs another particle. The architecture of the Prong CVN network closely follows the GoogLeNet [38] network with minor modifications.

The Prong CVN approach gives a better performance than the Log-Likelihood approach, and it allows NOvA to reuse the well-researched computer vision techniques, instead of developing new solutions. On the other hand, evaluation and training of the convolutional neural networks are computationally intensive tasks and they significantly impact the overall reconstruction speed.

5.4 Event Identification

In the previous sections of this chapter, I have described the basic NOvA reconstruction algorithms that allow us to identify clusters of hits belonging to the individual

particles, reconstruct particle tracks, and infer information about what kind of particle created such clusters. Using these basic reconstruction objects we can build more complex algorithms that will output objects directly useful for the neutrino oscillation analysis.

The first type of information that we need for the analysis is what kind of neutrino interaction (or cosmic ray) produced particles in a given slice. Let us review for a moment the types of neutrino interactions that are relevant for the analysis.

As was discussed in section 2.1, neutrinos can interact via Charged Current (CC) interaction (mediated by $W^{\pm}$ bosons) or via Neutral Current (NC) interaction (mediated by Z bosons) with the matter. In the Neutral Current interactions the incoming neutrino exchanges momentum with the detector matter via Z bosons, but the neutrino itself remains unaffected and is present in the final state particles. Since neutrinos do not have an electric charge they travel through the detector medium without depositing light.

This means that in the NC interactions the incoming neutrino eventually leaves the detector carrying away some fraction of its initial energy. Since we cannot see a track of the leaving neutrino, it is impossible to estimate the amount of energy this neutrino carried away. And this in turn makes it impossible to estimate the initial energy of the neutrino. Furthermore, the situation is complicated by the fact that different neutrino flavors ν_e, ν_μ and ν_τ behave in the same way in the NC interactions, which makes it impossible to distinguish neutrino flavors of the NC events.

Therefore, for the purposes of the oscillation analysis, we would like to identify events that look like NC events and treat them as background.

On the other hand, the Charged Current (CC) interactions are mediated by $W^{\pm}$ bosons. In the CC interaction, a neutrino is transformed into a lepton of the corresponding flavor $\nu_X \rightarrow X + W^{+}$. Using the final state lepton it is possible to predict the flavor of the incoming neutrino. Additionally, if the final products of the neutrino interaction stop within the detector volume we can make estimates about the initial energy of the incoming neutrino.

Consequently, the oscillation analysis is primarily interested in the identification of the Charged Current events and in determining their flavor and energy.

In this section, we will discuss methods that NOvA uses to separate Charged

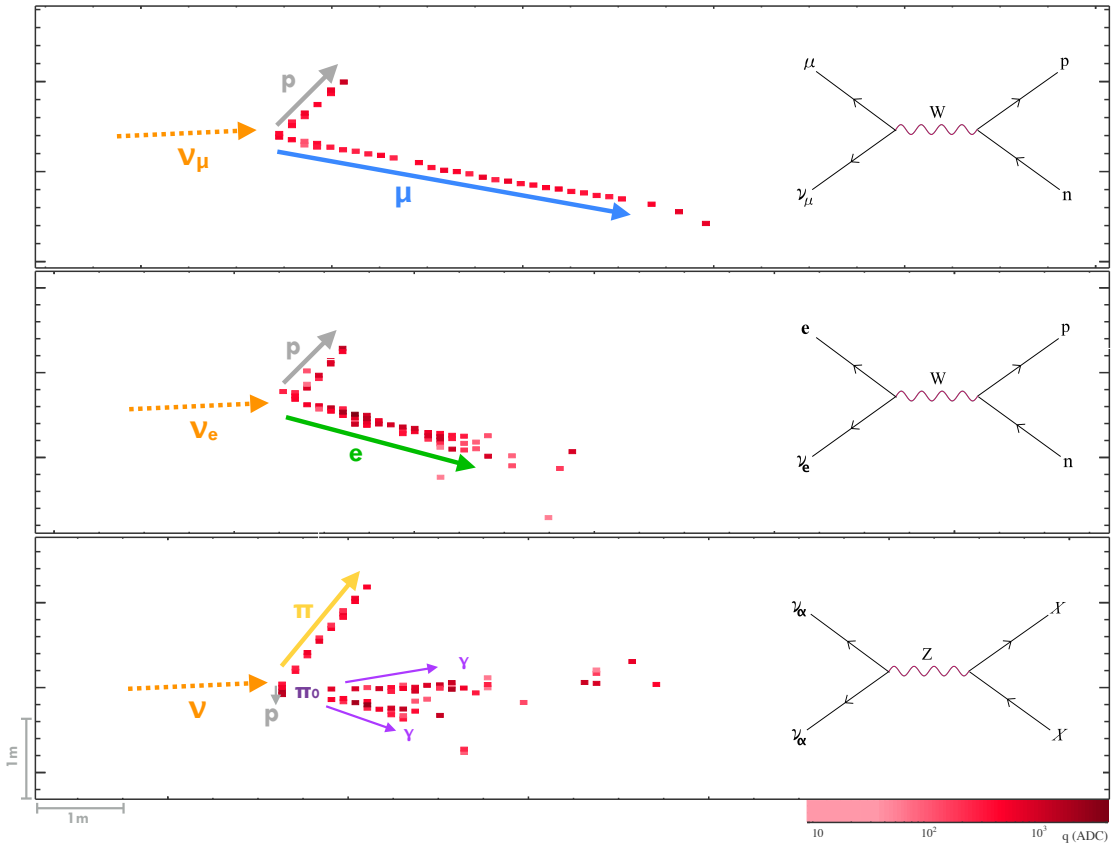


Figure 5.1: Examples of neutrino interaction event topologies observed in the NOvA detectors. The top pane depicts a NuMu-CC interaction. The middle one shows a NuE-CC interaction. The bottom pane illustrates an NC interaction.

Current events from Neutral Current events and from cosmic rays.

5.4.1 NuMu CC Events Identification with RemID

In the Charged Current interactions of the muon neutrinos ν_μ, the initial neutrino undergoes conversion into a muon and the remaining energy produces hadronic activity near the interaction vertex. As was discussed in subsection 5.3.1, the resulting muons leave distinctive long narrow tracks. There is no other probable interaction in the detector that can produce muons (apart from cosmic rays, but we will talk about them in a separate section). Therefore, if we can detect a muon in the neutrino interaction, then such interaction is a NuMu CC interaction.

We already have an algorithm RemID that can predict whether a given track was produced by a muon. This algorithm can be used for the purposes of the event identification as well. To get a numeric estimate of the probability that a given neutrino event is a NuMu CC one, we can just use the RemID value of the track with the highest RemID score in this event. If the RemID score of the event is above some threshold (to be determined at the Analysis stage) then we will predict that this event is a NuMu CC event.

5.4.2 NuE CC Events Identification with ShowerLID

Similar to how we reused RemID designed to detect muons in order to identify NuMu CC events, we can reuse ShowerLID designed to identify electron showers in order to identify NuE CC events. This NuE CC classifier looks at the most energetic shower in the event and tries to predict whether this shower was produced by an electron. Such prediction is performed with the help of a simple neural network having a Multilayer Perceptron (MLP) architecture [36]. Apart from the Log-Likelihoods of the most energetic shower predicted by ShowerLID, this network also uses several additional inputs:

- Shower Energy Fraction – shower energy divided by the total energy of the slice.
- Vertex Energy – total energy within 8 planes close to the interaction vertex.
- Shower Gap – a distance between the interaction vertex and the beginning of the shower.
- $\cos\theta$ – angle that the shower makes with the beam direction.
- π^0 Invariant Mass. The biggest backgrounds to the electron shower identification are the showers produced by photons (originating from pion decays $\pi^0 \to \gamma + \gamma$). Using the assumption that the leading shower in the slice is produced by one of such photons, the algorithm iterates over all other showers and uses their energies and directions in order to calculate the pairwise invariant

mass with the leading shower. The value of the invariant mass that is closest to the neutral pion mass m_{π^0} is used as another input variable.

The resulting MLP classifier is trained to differentiate whether the leading shower in the event was produced by an electron vs neutral pion. If MLP predicts (the threshold for such prediction is established at the Analysis stage) that the most energetic shower in the event is produced by an electron then we infer that the interaction is a NuE CC interaction.

5.4.3 Multiclass Event Identification with Event CVN

Similar to how convolutional neural network was used for the particle type identification in subsection 5.3.3, it can be used for the event identification as well. Event CVN [39] (or simply CVN at NOvA) is a convolutional neural network that looks at the images of the events and tries to predict the type of the event (NC vs NuMu CC vs NuE CC vs NuTau CC vs Cosmic). It uses an architecture similar to Prong CVN and has similar costs and benefits as the Prong CVN algorithm

5.4.4 Cosmic Rays Rejection

Cosmic rays produce significant background at the Far Detector. If left unchecked it will make any analysis impossible. Fortunately, most of the cosmic rays are traveling from top to bottom of the detector, and pass through the detector. Therefore, simply requiring that the event hits should be fully contained in the detector volume (minus margin) will remove a large fraction ($\sim$ 98%) of the rays that do not stop in the detector.

In order to reduce background from the remaining cosmic rays, NOvA uses a special cosmic rejection algorithm [40]. This algorithm looks at the primary Kalman track in the event and tries to predict whether this track was caused by a cosmic ray or not. It relies on an AdaBoosted forest of decision tree classifiers [41] to make an inference about whether a given event is a cosmic ray.

In order to make such inference, the AdaBoosted forest of decision trees looks at the direction of the primary Kalman track, its length, height of the endpoints,

fraction of hits belonging to the track over the total number of hits in the event, and finally the output of the Event CVN classifier.

Using these inputs the cosmic rejection algorithm was trained to differentiate between cosmic rays and muon tracks produced in the NuMu CC interactions since these are the only tracks that may look like cosmic rays. As output, the cosmic rejection algorithm predicts a score indicating how likely a given event is a cosmic ray. If this score is above some threshold (to be determined at the Analysis stage) then the event is labeled as cosmic and excluded from the analysis.

5.5 Energy Reconstruction

The last piece of information required for the oscillation analysis is the energy of neutrinos that interact in the NOvA detectors. Since this document primarily focuses on the muon neutrino disappearance and electron neutrino appearance analyses we will focus here on the reconstruction of the ν_e and ν_μ energies.

As was noted above, the only way we can reliably detect a neutrino of a certain flavor is if that neutrino interacted with the detector via Charged Current interaction. Therefore, energy reconstruction is done only for the NuMu CC and NuE CC interactions.

For the energy reconstruction of such events to be complete, all particles produced by a neutrino interaction must stop inside of the detector. Therefore, we will only look at the events that are contained in the detector.

5.5.1 NuMu Energy Reconstruction

We begin our discussion of the energy reconstruction techniques at NOvA with the energy reconstruction of the ν_μ neutrinos interacting via Charged Current interactions. As was discussed in the event identification sections, such events have a distinctive muon track and some hadronic activity near the vertex. One way to reconstruct the original energy of neutrino might be to sum all calorimetric energies of the daughter particles and try to find a relationship between the total calorimetric energy and the true energy of the incoming neutrino. While such an approach can

be used, it gives a suboptimal energy reconstruction performance and the predicted energy is too sensitive to the calorimetric energy uncertainties.

A better approach to reconstruct energies of the NuMu CC events is to notice that muons deposit energy at an approximately constant rate as they travel through the detector. Therefore it may be possible to reconstruct muon energy from the muon track length [42]. Once the muon energy is reconstructed, the remaining hadronic energy part is inferred from the calorimetric energy of the hadronic activity.

In order to reconstruct the muon energy part of the NuMu CC interaction, a histogram of the true muon energy vs muon track length is made. Then, a piecewise linear spline function is fit to the peaks of this histogram (Figure 5.2). The fitted spline function can be used to estimate energy of a muon from its track length.

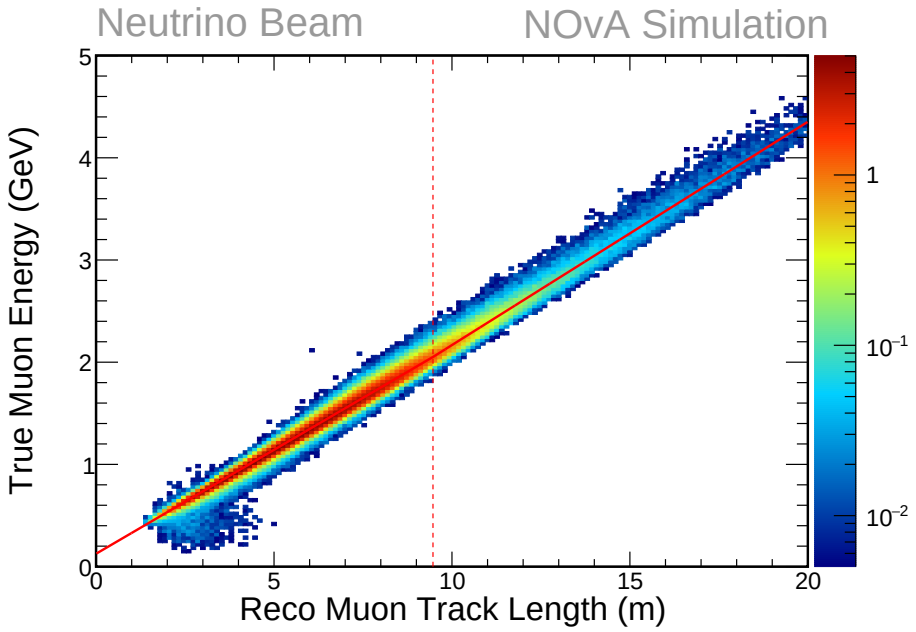


Figure 5.2: Histogram of the true muon energy vs muon track length for the primary muon in the NuMu-CC events. The red line shows a piecewise linear fit to the histogram.

The hadronic energy part is reconstructed in a similar way. A histogram of the true hadronic energy vs visible calorimetric energy is made. Then, peaks of this histogram are fit by another piecewise linear spline function (Figure 5.3). The

resulting fit is used to estimate hadronic energy from the visible calorimetric energy.

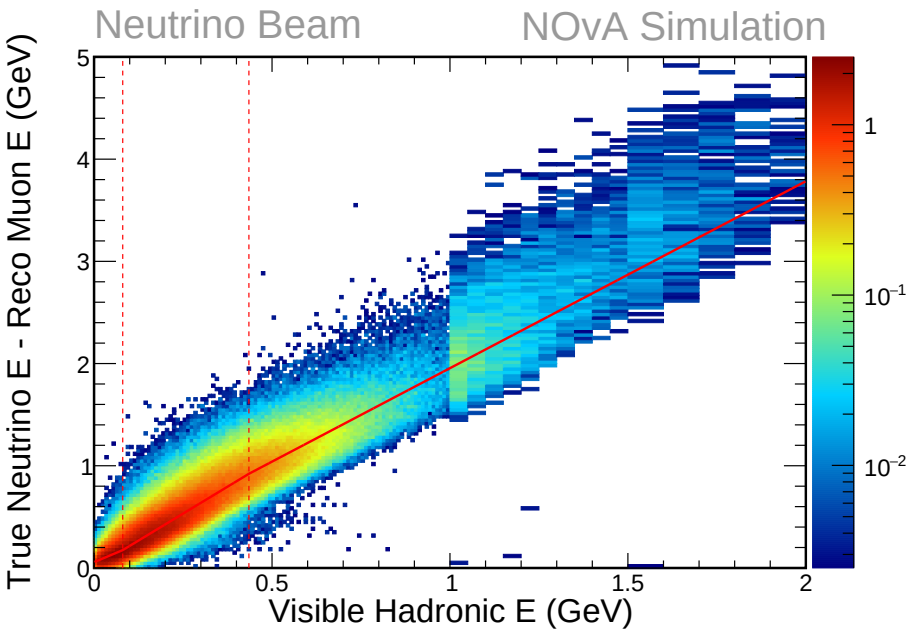


Figure 5.3: Histogram of the true neutrino energy minus reconstructed muon energy vs visible hadronic calorimetric energy for the NuMu-CC events. The red line shows a piecewise linear fit to the histogram.

Special care is taken to ensure proper energy reconstruction at the Near Detector. The NOvA Near Detector has a muon catcher region which contains a collection of steel plates interspersed with active detector planes. If we detect that the muon track enters the catcher region then we divide the muon track into two parts. One part is in the active detector volume and another in the muon catcher. We estimate energies of two parts separately using a special fit made for the muon catcher [42].

Comparing energy histograms for the hadronic energy (Figure 5.3) and the muon energy (Figure 5.2) parts we can see that the large fraction of the hadronic energy remains unexplained by the piecewise linear fit (compared to muon energy). This indicates a possibility that the multivariate reconstruction may be needed to improve the hadronic energy reconstruction performance.

The issue of the relatively poor hadronic energy reconstruction is alleviated by the fact that in the events that are used in the NOvA analysis 2/3 of the neutrino

energy comes from the energy of the muon.

5.5.2 NuE Energy Reconstruction

In the previous section, we considered ν_μ neutrino energy estimation in Charged Current interactions. The ν_μ energy estimator heavily relied on the fact that the primary daughter muon leaves a long and narrow track, and deposits energy at an approximately constant rate. Unfortunately, in the ν_e Charged Current interactions, the primary daughter electron creates a wide electro-magnetic shower and there is no simple relationship between the dimensions of the shower and the energy of the electron.

In order to estimate the energy of the incoming ν_e in the Charged Current interactions, the NOvA NuE energy reconstruction algorithm relies on an observation that the detector response is different for the electro-magnetic and hadronic energy depositions [43]. Therefore, the NuE energy estimator tries to separate clusters of hits in each event into two categories: EM clusters and clusters produced by the hadronic activity. Then, it uses a polynomial fit to predict the ν_e energy based on the total calorimetric energies of the EM and hadronic clusters.

The cluster separation into EM and hadronic categories is performed by looking at the scores predicted by the Prong CVN algorithm. The NuE energy reconstruction algorithm calculates an EM score from the Prong CVN scores (5.9). If P_{EM} is greater than 0.5 then the cluster is classified as of EM origin and of hadronic origin otherwise.

$$P_{\text{EM}} = P_e + P_\gamma + P_{\pi^0} \tag{5.9}$$

Once clusters are classified into EM and hadronic categories, the energy estimation algorithm calculates the total EM calorimetric energy of the EM clusters. This energy is found by summing renormalized calorimetric energies of hits that the ShowerLID algorithm produced for the EM clusters. The hadronic calorimetric energy component is defined simply as the total calorimetric energy of the event minus the EM calorimetric energy part.

Figure 5.4 illustrates how the true neutrino energy depends on the EM and hadronic calorimetric energy parts. The second-order polynomial fit (5.10) is per-

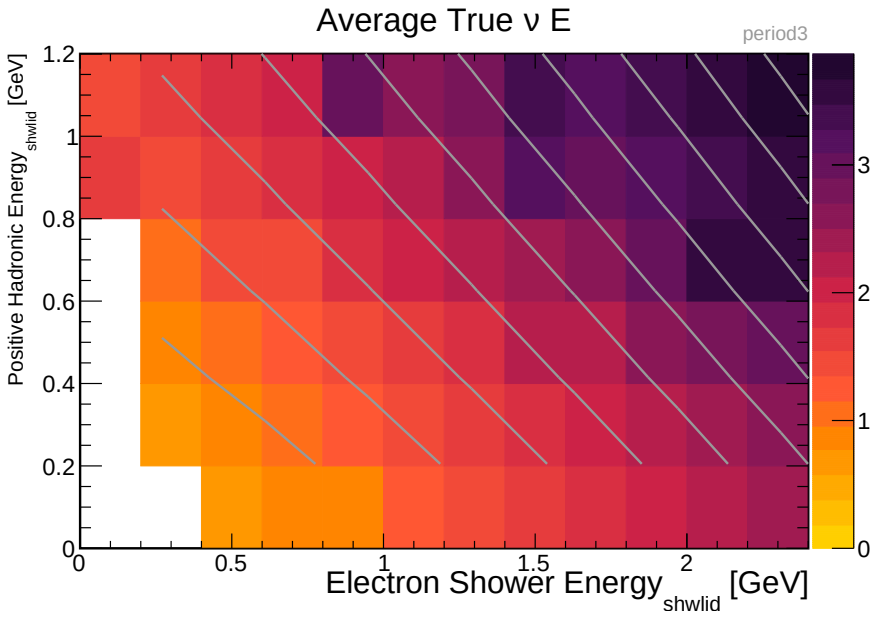


Figure 5.4: True ν_e energy distribution (color) plotted vs total calorimetric energy of the EM prongs (x-axis) and total calorimetric energy of the hadronic prongs (y-axis).

formed to the histogram shown in Figure 5.4.

$$E_{\text{reco}} = \alpha_1 E_{\text{EM}} + \beta_1 E_{\text{HAD}} + \alpha_2 E_{\text{EM}}^2 + \beta_2 E_{\text{HAD}}^2 \tag{5.10}$$

where $\alpha_{1,2}$, $\beta_{1,2}$ are the fit coefficients. This fit alone can be used to reconstruct ν_e energies, however, it turns out that the predicted energies are biased by about 6% [43]. In order to remove this bias, the energy reconstruction algorithm does the final step and rescales the predicted energy by $\sim \frac{1}{1+0.06}$.

5.5.3 Performance of Energy Estimators

In concluding the discussion of energy estimation methods at NOvA I would like to briefly touch on how the performance of energy estimators is assessed. This topic is relevant for the data analysis and development of better energy estimators.

The performance of energy estimators at NOvA is assessed by looking at the energy resolution which is defined as a histogram of the ratio of (Reco Energy − True Energy) to True Energy. The better the energy reconstruction the narrower

this histogram is. As a measure of narrowness we can use for example the RMS value of the energy resolution histogram, or we can fit a gaussian curve to the histogram and look at the spread of the gaussian curve σ.

For example, Figure 5.5 shows muon energy resolution predicted by the standard NOvA ν_μ energy estimator. In terms of RMS, its energy resolution is 5.1%. On the other hand, Figure 5.6 shows the energy resolution of the hadronic activity in the ν_μ-CC events. The RMS of the hadronic energy resolution is 30% which is much higher than the RMS of the muon energy 5.1%.

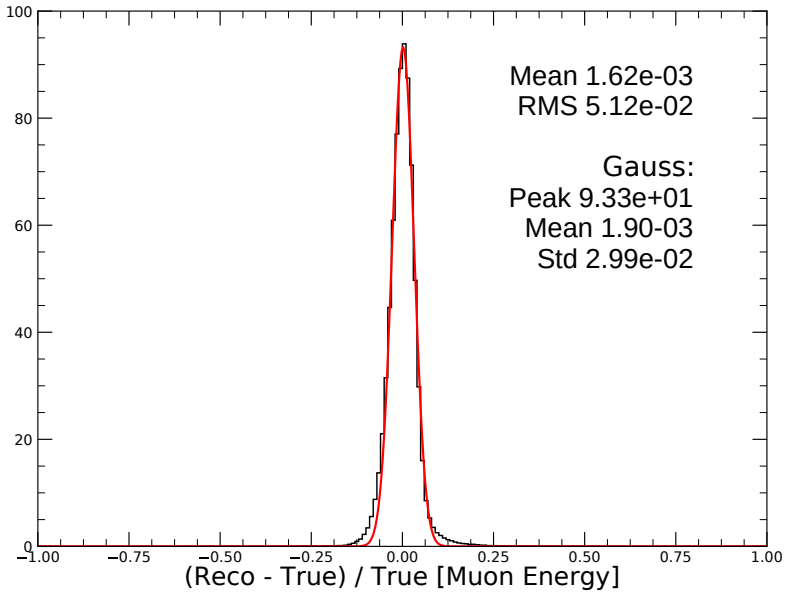


Figure 5.5: Muon energy resolution histogram for the standard NOvA ν_μ energy estimator.

The large difference in the RMS values of muon and hadronic energy resolutions illustrates the previous point that the large fraction of hadronic energy remains unexplained by the standard NOvA ν_μ energy estimator.

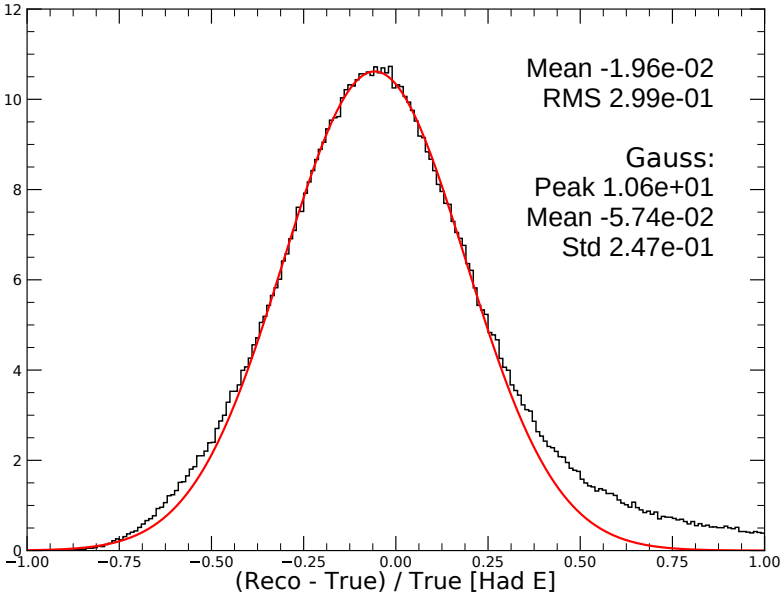


Figure 5.6: Hadronic energy resolution histogram for the standard NOvA ν_μ energy estimator.

Chapter 6

Analysis

In the previous chapters, we have looked at the design of the NOvA experiment, the calibration procedure of the detectors, and the reconstruction techniques used to identify interacting particles in the detector and estimate their energies. In this chapter, we will consider how we can use the reconstructed particle interactions to make an inference about the values of the neutrino oscillation parameters.

The process of data analysis has several stages. First, the activity in the detectors is separated into different categories (like NuMu CC, NuE CC, etc) in a process known as Decomposition. The decomposition is done by splitting samples (a process also known as making cuts) by the PID scores predicted by the event identification algorithms. At this stage, samples that are too contaminated by backgrounds or that have a poor reconstruction performance are discarded.

Then, the prediction is made about the rate of the Far Detector neutrino activity from the decomposed Near Detector sample. This process is called Extrapolation, and it uses the neutrino oscillation model to predict the neutrino sample composition at the Far Detector. Since the Extrapolation process relies on the neutrino oscillation model, the final prediction is dependent on the oscillation parameters. In order to apply the neutrino oscillation model, we need to know energies of the neutrinos at the Near Detector, which are computed by using energy estimation algorithms discussed in section 5.5.

Once we have obtained the prediction about the neutrino activity in the Far Detector from the Near Detector sample we compare this prediction to the actual

activity in the Far Detector. Since the prediction about the Far Detector activity depends on the values of the neutrino oscillation parameters, we can obtain different predictions by varying their values. In this process, we may observe that some combination of the neutrino oscillation parameters gives the best agreement between the predicted Far Detector neutrino activity and the observed one. The goal of the NOvA neutrino oscillation analysis is to find such a set of the neutrino oscillation parameters, that gives the best agreement between the predicted and the actual Far Detector activity.

After the best values of the neutrino oscillation parameters are found we need to estimate uncertainties of these parameters. The uncertainties of the oscillation parameters have two parts – statistical and systematic. The statistical uncertainty can be easily estimated using the standard statistical methods, while the systematic uncertainties are specific to each experiment and require careful study of different NOvA features.

In this chapter, we will discuss sample Decomposition, Near Detector data Extrapolation, finding the best oscillation parameters, and estimation of their uncertainties.

6.1 Sample Selection

The goal of the sample selection process is to select samples of NuMu CC and NuE CC events that have good reconstruction properties and contain minimal contamination from the background events [44]. There are three types of cuts that are used to obtain the said samples: quality cuts, containment cuts, and PID cuts.

The quality cuts try to reject events with the most obvious errors in reconstruction. For example, they reject events in which the reconstruction algorithm failed to identify an interaction vertex, or events without any prongs, events that have too many or too few hits. For selecting the NuMu CC sample, quality cuts require the presence of a reconstructed Kalman track and require the reconstructed energy to be in the range from 0 to 5 GeV. For the NuE CC sample, quality cuts additionally require the presence of a shower reconstructed by ShowerLID.

The quality cuts filter out only the most basic reconstruction failures. However,

as I have described before, for the analysis purposes we need to know the energy of the interacting neutrinos. In order to get a reasonable estimate of the energy of the neutrino, we need a sample where none of the neutrino daughter particles left the detector since they can carry an undetermined amount of energy with them. That is why the containment cut is applied – it ensures that all neutrino daughter particles have stopped in the detector, which makes energy reconstruction much more accurate. The fact that all daughter particles are contained in the detector volume also improves the performance of the event identification algorithms.

In essence, the containment cuts require that all prongs in the event are at least at some finite distance (D) from the detector edges. This distance is tuned to make sure that the probability of an event passing the containment cut, but still having particles that escape the detector is small. Unfortunately, tuning this metric alone will tend to make the distance $D \to \infty$, i.e. by simply rejecting all events, we can guarantee that no event will have particles that escape the detector. Therefore, additional metrics are used, like the fraction of signal events to the background, to make sure that we do not simply reject all events.

The containment distance is determined separately for different detectors (ND and FD) and different detector edges. Additional cuts are made for the NuMu and NuE CC samples that take into account properties specific to these event types. For example, for the NuMu CC events, we require that only one Kalman track enters the Muon Catcher in the Near Detector. On the other hand, for the NuE CC, we require that no activity occurs in the Muon Catcher of the Near Detector.

The containment cuts provide an additional benefit of removing cosmic muons at the Far Detector and muons originating in rock in the Near Detector. This happens because cosmic rays and rock muons originate outside of the detector and enter it through one of its edges. Therefore, requiring that hits should be contained at some distance from the detector edges removes such muons.

After the most obvious reconstruction failures have been removed and the sample was cleaned of events that have escaping particles, we can split the sample into finer subsamples relevant for the NuMu and NuE analyses.

6.1.1 NuMu CC Sample Selection

To select the sample of ν_μ-CC neutrino interactions for the muon neutrino disappearance analysis, first NOvA removes events that have muon-like tracks that are not aligned with the beam direction. This is achieved by making cuts on the angle that muon tracks make with the beam direction. Such cuts reject backward going cosmic muons.

Then, the three-dimensional distribution of events in a space of RemID values, Cosmic Rejection scores, and NuMu CC CVN values is considered. A rectangular cut is found (via the grid search) in this three-dimensional space that maximizes the Figure of Merit FOM $= \frac{S^2}{S+B}$ where S is the number of selected signal events and B is the number of selected background events. This three-dimensional cut is used to select the sample of NuMu CC events for analysis.

6.1.2 NuE CC Sample Selection

The NuE CC sample that is used for the analysis has two major parts – the Core sample and the Peripheral sample. The Core NuE CC sample has the least amount of background contamination and good energy reconstruction properties. The Peripheral sample contains events that have relatively good reconstruction properties, but not pure enough for the main analysis. Both samples have different selection rules.

NuE CC Core Sample Selection

The selection of the NuE CC events is performed with a help of several cuts. First, the cosmic background is reduced by performing cuts on the value of the transverse (to the beam direction) momentum. These cuts reject cosmic rays that are not aligned with the beam direction. An additional cut called the Backward Photon cut is applied to the back part of the detector. This cut rejects cosmic photons that might arrive from the outside of the detector by requiring that the photon showers close to the back of the detector be directed outwards.

After the cosmic rejection cuts, the main selection rules are applied. First, several preselection criteria are used, requiring that the energy, number of hits, and the size

of the event be in a reasonable range for a NuE CC event. Then, the main selection cut is performed on the value of the NuE CC CVN score, by requiring that the NuE CC CVN score is above 0.75. This cut was found by optimizing the Figure of Merit metric FOM = $\frac{S^2}{S+B}$ where S is the number of selected signal NuE CC events and B is the number of background events.

NuE CC Peripheral Sample Selection

The less pure sample of NuE CC events is constructed by considering events on the periphery of the core sample that fail some of the core sample cuts (either containment or cosmic rejection or preselection). The events failing either of those cuts, but still having a very high value of the NuE CC CVN score (> 0.99), or having both a high value of the Nue CC CVN score (> 0.95) and high value of the Cosmic Rejection score are included in the peripheral NuE sample.

6.2 Binning Selection

The selected NuMu and NuE CC samples in the previous section have low background contamination but still are made of many different events with different energies and interaction properties. However, not all selected events are equally important for the analysis. For example, the selected NuMu CC sample has events with energies from 0 to 5 GeV, but for the analysis, we are mostly interested in a region around 2 GeV where the neutrino oscillation phenomenon occurs. Therefore, it would make sense to bin the NuMu CC sample by energy more finely around 2 GeV where more events are available and the majority of the interesting physics is concentrated. In this section, I will discuss how the analysis samples are binned and further subdivided to get better measurements of the neutrino oscillation parameters.

6.2.1 NuMu CC Sample Binning

As mentioned above, for the analysis we are mostly interested in the events with energies around 2 GeV where neutrino oscillations occur and the neutrino flux is maximal. Therefore, the NuMu CC sample is binned in the reconstructed energy

variable with fine binning between $1-2$ GeV and progressively coarse binning outside of that interval. The bin edges were chosen in this way since it maximizes the precision of measurement of the neutrino oscillation parameters.

Additionally, the NuMu CC analysis sample is made of neutrino interactions of different types. For example, Figure 6.1 demonstrates the energy distribution of the NuMu CC sample, separated by the neutrino interaction mode (QE – quasi-elastic events; RES – resonant scattering events; DIS – deep inelastic scattering events; MEC – meson exchange current events). These neutrino interaction modes can be correlated with the average amount of energy in a NuMu CC neutrino interaction ($\nu_\mu \to \mu + \text{hadrons}$) that is transferred to the hadronic activity.

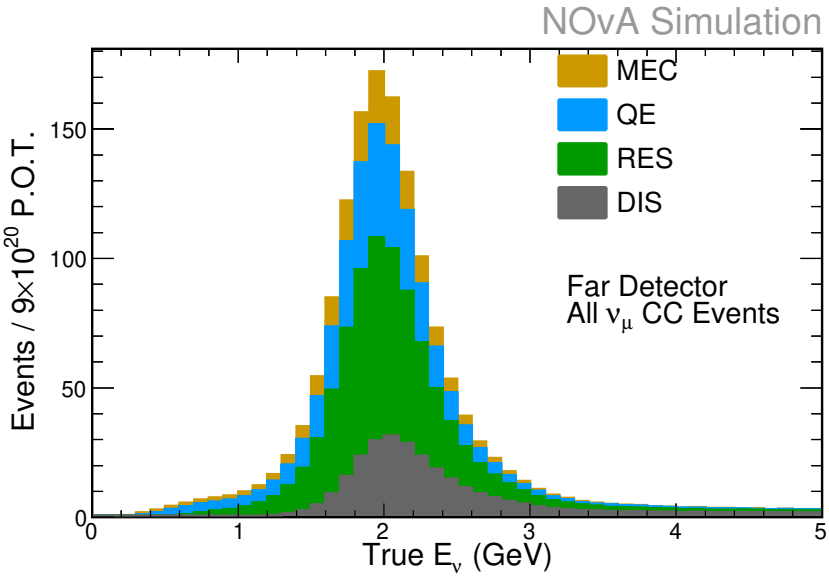


Figure 6.1: Simulated (unoscillated) true energy distribution of the ν_μ-CC events in the NOvA Far Detector split by the interaction modes.

For example, in the quasi-elastic (QE) NuMu CC neutrino interaction, most of the neutrino energy is transferred to the primary muon. On the other hand, in the deep inelastic scattering mode (DIS) a significant fraction of the neutrino energy goes into the hadronic activity. This becomes important since as we have seen in subsection 5.5.1 the neutrino energy reconstruction performance is good only for events where most of the neutrino energy goes into a primary muon (as opposed to

a hadronic activity). In other words, the standard NuMu CC energy estimator has a much better performance on the QE events compared to the DIS events.

The NOvA analysis tries to take advantage of the superior neutrino energy reconstruction for the QE events and RES events, compared to the DIS events, by subdividing the selected NuMu CC analysis sample into 4 bins, depending on the fraction of neutrino energy that goes into the hadronic activity. Figure 6.2 shows a two-dimensional histogram of hadronic energy fraction (E_{had}/E_{ν}) vs neutrino energy that is used to subdivide the analysis sample into 4 parts (quartiles), each having the same number of neutrino events.

The NuMu CC events in the first hadronic energy quartile are the events where the majority of the neutrino energy goes into the primary muon. These events have a very small energy reconstruction error. Events in the fourth hadronic energy quartile have the majority of the neutrino energy converted into the hadronic activity and correspondingly have a very large error of the reconstructed energy. The four hadronic energy fraction quartiles are analyzed separately in order to exploit the low energy error of the first quartiles for a more precise oscillation parameter estimation.

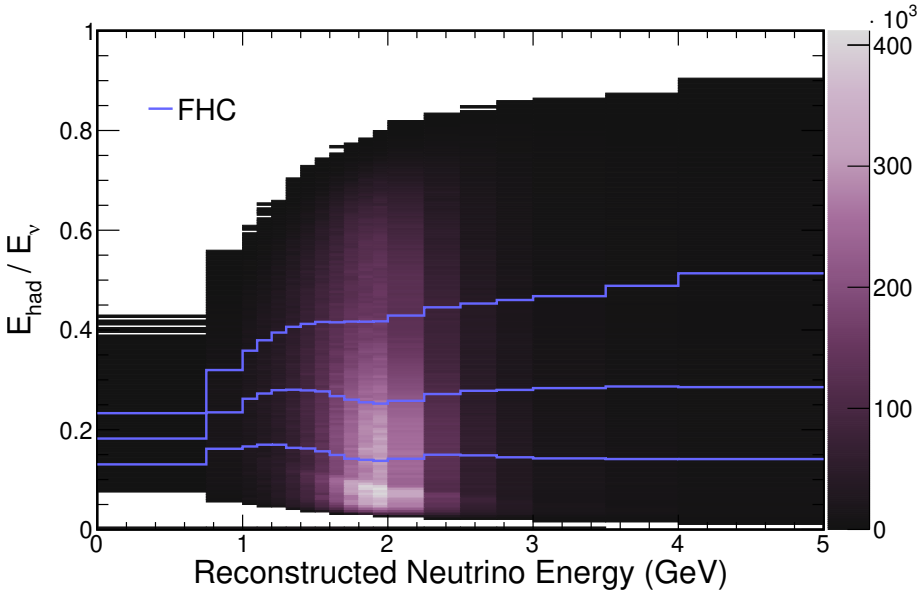


Figure 6.2: Two-dimensional histogram of Hadronic Energy Fraction vs Reconstructed Neutrino Energy for the Far Detector FHC NuMu CC sample. The three blue lines indicate hadronic energy fraction quartile boundaries. Source [45].

6.2.2 NuE CC Sample Binning

The NuE CC sample has two parts: Core and Peripheral samples. The Peripheral sample has large background contamination and bad energy reconstruction properties. Therefore, it was decided to use just a single bin for the entire Peripheral sample [46]. On the other hand, the structure of the Core sample is much richer. Similar to the partitioning of the NuMu CC sample into various hadronic energy fraction bins, the NuE CC Core sample is divided into two bins based on the value of the NuE CC CVN score (or CVNe score for short). The boundaries of the CVNe bins were found by maximizing the Figure of Merit

$$\text{FOM} = \sum_{i=1}^{2} \left(\frac{S_i^2}{S_i + B_i} \right) \tag{6.1}$$

where the sum over i is the sum over the CVNe bins, S_i is the number of the NuE CC events in a CVNe bin number i, and B_i is the number of the background events in a bin i. The two CVNe bins of the Core sample are further binned in the reconstructed neutrino energy in range $1 - 4\,\text{GeV}$. Figure 6.3 shows an example of the binning of the NuE CC sample into Core/Peripheral parts, Low/High CVNe components, and reconstructed energy bins.

6.3 Decomposition

Using the selected NuMu CC and NuE CC samples from the Near Detector data we would like to obtain the predicted NuMu CC and NuE CC samples at the Far Detector with the help of the neutrino oscillation model. However, the NuMu CC and NuE CC samples have some background contamination. The background contamination makes it impossible to directly apply the neutrino oscillation model to the entire sample. Therefore, before the oscillation model can be applied, the NuMu CC and NuE CC analysis samples are decomposed into various components (e.g. ν_μ-CC events, $\bar{\nu}_\mu$-CC, ν_e-CC, ν_τ-CC, NC, Cosmics, etc). In this section, I will describe how the sample decomposition is performed.

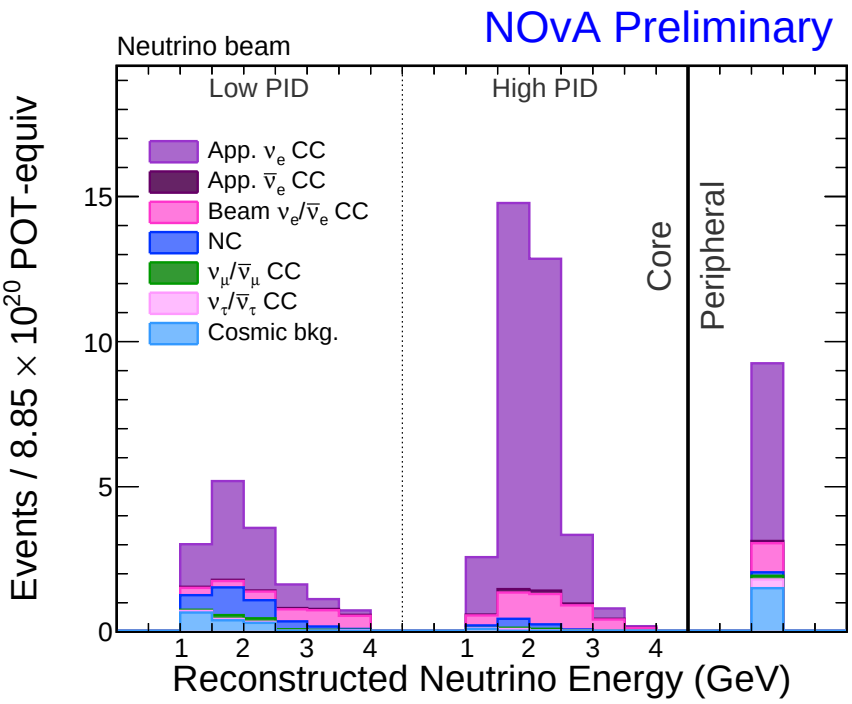


Figure 6.3: Example of a NuE CC sample binning at the Far Detector. The sample is made of two parts – the Core sample (left) and the Peripheral sample (right). The Core sample is subdivided into two parts (Low PID, High PID) based on the value of the NuE CVN score. Each part of the Core sample is further binned in the reconstructed neutrino energy between $1 - 4\,\text{GeV}$.

6.3.1 NuMu CC Sample Decomposition

The NuMu CC sample is used for the ν_μ disappearance analysis. The ν_μ disappearance analysis counts how many ν_μ neutrinos have disappeared from the beam due to the neutrino oscillations, while the beam was traveling from the Near Detector to the Far Detector. For the ν_μ disappearance analysis, we assume that the MC correctly predicts background event rates at the Far Detector. In other words, the extrapolation procedure only extrapolates ν_μ-CC and $\bar{\nu}_\mu$-CC components of the data from the Near Detector to the Far Detector. Then, it adds the simulated background components at the Far Detector to obtain the predicted FD sample.

Therefore, the NuMu CC sample decomposition only tries to decompose the Near Detector data into ν_μ-CC and $\bar{\nu}_\mu$-CC parts. In order to perform such a decomposi-

tion, NOvA heavily relies on the Near Detector MC. It assumes that the MC correctly models the fraction of background interactions at the ND. Under this assumption, a fraction of the simulated background interactions is subtracted from the Data:

$$S_{Data} = N_{Data} - B_{MC} \tag{6.2}$$

where S is the number of signal events (ν_μ and $\bar{\nu}_\mu$-CC), B is the number of background events (everything that is not ν_μ and $\bar{\nu}_\mu$-CC) and N is the total number of events. Such a subtraction is performed for each hadronic energy fraction quartile and reconstructed energy bin (as discussed in subsection 6.2.1).

Next, the total number of signal events S_{Data} is further decomposed into the ν_μ-CC and $\bar{\nu}_\mu$-CC components. To perform this decomposition NOvA relies on the Near Detector MC and assumes that the MC correctly models the fractions of the ν_μ-CC to the $\bar{\nu}_\mu$-CC events:

$$\begin{aligned} S_{Data}[\nu_\mu] &= S_{Data} \frac{S_{MC}[\nu_\mu]}{S_{MC}[\nu_\mu] + S_{MC}[\bar{\nu}_\mu]} \\ S_{Data}[\bar{\nu}_\mu] &= S_{Data} \frac{S_{MC}[\bar{\nu}_\mu]}{S_{MC}[\nu_\mu] + S_{MC}[\bar{\nu}_\mu]} \end{aligned} \tag{6.3}$$

This decomposition is performed for each NuMu CC bin. The resulting signal components $S_{Data}[\nu_\mu]$ and $S_{Data}[\bar{\nu}_\mu]$ are used in the extrapolation procedure for the ν_μ disappearance analysis.

6.3.2 NuE CC Sample Decomposition

The decomposition of the NuE CC sample follows the NuMu CC sample scheme, but it is slightly more complicated. Similar to the NuMu CC case it assumes that at the Near Detector we can correctly simulate relative ratios of different interaction types. However, while the NuMu CC decomposition assumed that only relative fractions of ν_μ and $\bar{\nu}_\mu$ events were simulated correctly, the NuE CC sample decomposition assumes that relative fractions of all interaction types (ν_μ-CC, ν_e-CC, NC, and their antiparticle variants) are simulated correctly.

Under such an assumption, the decomposition of the NuE CC sample into com-

ponents is achieved as:

$$\mathrm{N}_{\mathrm{Data}}[x] = \mathrm{N}_{\mathrm{Data}} \frac{\mathrm{N}_{\mathrm{MC}}[x]}{\mathrm{N}_{\mathrm{MC}}} \tag{6.4}$$

where $\mathrm{N}_{\mathrm{Data}}$ and N_{MC} are the total number of events in Data and MC correspondingly, $\mathrm{N}_{\mathrm{Data}}[x]$ and $\mathrm{N}_{\mathrm{MC}}[x]$ are the numbers of events of type x (ν_μ-CC, ν_e-CC, NC, etc). As in the NuMu CC case, this decomposition is performed per NuE CC sample type (Core High PID, Core Low PID, Peripheral) and per reconstructed energy bin Figure 6.3.

The NuE CC sample decomposition outlined above relies heavily on the assumption that the simulation of relative fractions of different interaction types is correct. The NOvA experiment has another NuE CC sample decomposition method (called BEN) that is data-driven and is capable of estimating the fraction of ν_e events produced in the beam [47].

The idea behind the BEN decomposition is to use the fact that the beam ν_e and ν_μ neutrinos come from the decays of charged pions and kaons in the decay pipe. If we can find the number of ν_μ-CC events in the beam (for example using the NuMu CC decomposition) then the number of parent pions and kaons in the decay pipe can be estimated. Then, using these estimated numbers, combined with the decay rates of pions and kaons, the number of ν_e in the beam can be predicted.

6.4 Extrapolation

The extrapolation procedure aims to obtain the prediction about the Far Detector neutrino event counts from a decomposed Near Detector sample. The extrapolation procedure differs between different analyses and neutrino interaction types. In particular, we perform the full extrapolation from the Near Detector event counts for some neutrino interactions and rely on the Far Detector MC to estimate counts of other neutrino interactions.

The decision on whether to use the full extrapolation or just rely on the Far Detector MC for a given neutrino interaction type is based on our expectation of the number of such interaction types at the Far Detector. If we believe that some neutrino interaction will be very rare at the Far Detector (like ν_τ-CC interactions)

then we can estimate their number from the Far Detector MC. On the other hand, if some other neutrino interaction type is relatively frequent, then we use the full extrapolation from the Near Detector data.

6.4.1 Full Extrapolation

The decision on which neutrino interaction types undergo the Full Extrapolation differs between the ν_μ disappearance and the ν_e appearance analyses. For the ν_μ disappearance analysis, the NuMu CC events (and Anti NuMu CC) are the largest contributor to the predicted neutrino spectrum at the Far Detector. The NuMu CC events at the Far Detector can be either produced by the beam ν_μ neutrinos that survived neutrino oscillation, or by the beam ν_e, ν_τ neutrinos that have oscillated into the ν_μ flavor. The latter part of the NuMu CC events is very small and therefore we use the Far Detector MC to estimate it. Only the NuMu CC events coming from the beam ν_μ neutrinos are predicted using the full extrapolation procedure.

For the purposes of the ν_e appearance analysis, signal events at the FD are the NuE CC events that come from the beam ν_μ that oscillate into the ν_e flavor. The Full Extrapolation in the ν_e appearance analysis case is performed only for such events.

The Full Extrapolation procedure consists of several steps. Figure 6.4 illustrates these extrapolation steps using the extrapolation of the NuMu CC ND events into the NuMu CC FD events as an example. It begins with considering a reconstructed neutrino energy spectrum of the NuMu CC component (obtained via decomposition of the ND Data) in the top left corner.

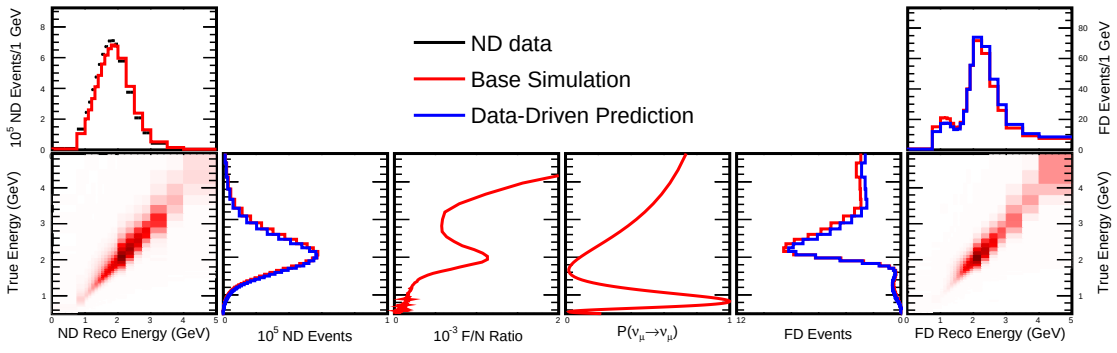


Figure 6.4: NuMu CC extrapolation diagram

Next, it maps the reconstructed energy spectrum at the ND into the spectrum of true energies using the simulated matrix (bottom left corner) of probabilities $P(E_{\text{reco}} \rightarrow E_{\text{true}})$. The resulting spectrum of true energies is depicted at the plot labeled "ND Events" in Figure 6.4.

Then, the actual extrapolation from the Near Detector true energy spectrum into the Far Detector energy spectrum is performed. This extrapolation relies on the kinematic characteristics of the beam (F/N ratio) and the neutrino oscillation model ($P(\nu_\mu \rightarrow \nu_\mu)$).

Finally, the extrapolation procedure maps the extrapolated true neutrino energy spectrum into the reconstructed neutrino energy spectrum using another simulated matrix (bottom right corner) of probabilities $P(E_{\text{true}} \rightarrow E_{\text{reco}})$. As the result of this procedure, we obtain the reconstructed energy spectrum of the NuMu CC events at the Far Detector in the top right corner of Figure 6.4.

6.4.2 Partial Extrapolation of Backgrounds

The ν_e appearance analysis uses samples with large contamination from backgrounds. These backgrounds include:

- NuE CC background interactions produced by the beam ν_e neutrinos that have survived oscillations.
- NuMu CC background interactions produced by the beam ν_μ neutrinos that survived oscillations.
- Neutral Current interactions.

In order to estimate the contributions of these backgrounds at the Far Detector, another type of data-driven extrapolation is used. In this extrapolation type, we rely on the Far Detector simulation of such backgrounds and reweigh the simulated Far Detector reconstructed energy spectrum by the ratio of Near Detector Data to the MC reconstructed energy spectra.

NonExtrapolation of Small Backgrounds

As was noted before, the analysis sample also has a small amount of contamination from other interaction types (e.g. ν_τ-CC interactions). Since this contamination is very small we simply rely on the Far Detector MC to estimate it, without performing any extrapolation from the Near Detector data.

6.5 Estimation of the Oscillation Parameters

In order to obtain an estimate of the neutrino oscillation parameters, the predicted Far Detector neutrino energy spectrum (obtained via extrapolation procedure) is compared to the measured spectrum of neutrino energies at the Far Detector. The degree of agreement between the actual and expected spectra is evaluated using the log-likelihood [49] of obtaining the actual spectrum assuming that it follows the Poisson distribution with means given by the predicted spectrum (6.5) (parametrized by the neutrino oscillation parameters $\vec{\theta}$).

$$-2\mathrm{LL}(\vec{\theta}) = -2\sum_{i=1}^{N}\left[N_i^{\mathrm{Pred}}(\vec{\theta}) - N_i^{\mathrm{Data}} + N_i^{\mathrm{Pred}}(\vec{\theta})\log\left(\frac{N_i^{\mathrm{Data}}}{N_i^{\mathrm{Pred}}(\vec{\theta})}\right)\right] \tag{6.5}$$

where multi-index i indicates the analysis bin (energy, quartile, etc), $N_i^{\mathrm{Pred}}(\vec{\theta})$ is the number of predicted neutrino events in bin i, and N_i^{Data} is the number of measured events in bin i.

The estimate of the neutrino oscillation parameters $\vec{\theta}_0$ is obtained by minimizing Equation 6.5 with respect to $\vec{\theta}$. In addition to finding an estimate of the oscillation parameters, we can obtain their confidence intervals. Under the assumption that N_i are large (Gaussian limit), the quantity $\Delta\chi^2 \equiv -2\left[\mathrm{LL}(\vec{\theta}) - \mathrm{LL}(\vec{\theta}_0)\right]$ follows a χ^2 distribution with n degrees of freedom, where n is equal to the number of varied oscillation parameters during minimization [51]. The χ^2 distribution can be used to obtain confidence intervals for the neutrino oscillation parameters. Figure 6.5 shows an example of such confidence interval for the NuMu disappearance analysis (also known as sensitivity contours) obtained via such a procedure.

Unfortunately, the confidence interval construction procedure above relies on the

Gaussian limit that is not generally applicable to the NOvA experiment. A better procedure exists for obtaining confidence intervals – the Feldman-Cousins procedure [50]. The Feldman-Cousins procedure runs a large number of mock experiments to faithfully simulate the probability distribution of Equation 6.5. This procedure, however, requires a lot of computational power and offers just small corrections to the contours obtained in the Gaussian limit. Therefore, in this thesis, I will focus just on the sensitivity contours obtained in the Gaussian limit.

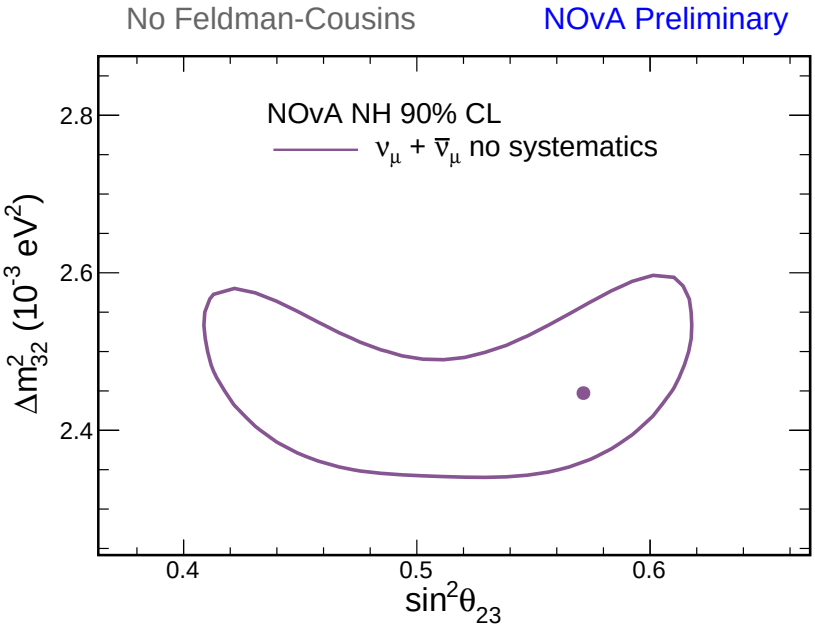


Figure 6.5: Example of a sensitivity contour in the Δm^2_{32} vs $\sin^2\theta_{23}$ plane for the NuMu disappearance analysis. The violet dot represents the best fit values of the oscillation parameters. The violet contour corresponds to the 90% confidence contour in the Gaussian limit.

6.6 Systematic Uncertainties

The oscillation parameter estimation procedure described in section 6.5 assumes that the only errors in the NOvA experiment are the statistical errors. Unfortunately, that is not the case and there are multiple systematic uncertainties at NOvA that we

should account for. The systematic uncertainties at NOvA can be roughly divided into several categories:

- Uncertainties of the NuMI beam flux and composition.
- Uncertainties of the cross-section of neutrino interactions with matter.
- Uncertainties of secondary particle transport through the detector.
- Uncertainties of the detector calibration.
- Uncertainties of the background contamination estimates.

The NOvA experiment spends considerable effort enumerating all possible systematic uncertainties and assessing their magnitude. In order to estimate the effect of the systematic uncertainties on the measurements of the neutrino oscillation parameters, spectra of predicted Far Detector neutrino energies are constructed as a function of the magnitudes of the systematic shifts δ_α (according to a procedure described below) where α is the name of a systematic shift.

The objective function that we used in the previous section (6.5) to find the estimates of the neutrino oscillation parameters is modified to include the systematic shifts and it becomes a function of the magnitudes of the systematic shifts δ_α (6.6).

$$\begin{aligned} -2\mathrm{LL}(\vec{\theta},\vec{\delta}) &= -2\mathrm{LL}_{\mathrm{Spectr}}(\vec{\theta},\vec{\delta}) - 2\mathrm{LL}_{\mathrm{Shifts}}(\vec{\delta}) \\ -2\mathrm{LL}_{\mathrm{Spectr}}(\vec{\theta},\vec{\delta}) &= -2\sum_{i=1}^{N}\left[N_i^{\mathrm{Pred}}(\vec{\theta},\vec{\delta}) - N_i^{\mathrm{Data}} + N_i^{\mathrm{Pred}}(\vec{\theta},\vec{\delta})\log\left(\frac{N_i^{\mathrm{Data}}}{N_i^{\mathrm{Pred}}(\vec{\theta},\vec{\delta})}\right)\right] \\ -2\mathrm{LL}_{\mathrm{Shifts}}(\vec{\delta}) &= \sum_{\alpha=0}^{M}\frac{\delta_\alpha^2}{\sigma_\alpha^2} \end{aligned} \tag{6.6}$$

where the second term $\mathrm{LL}_{\mathrm{Shifts}}(\vec{\delta})$ is the log-likelihood associated with the systematic shift δ_α, assuming that it follows the Normal distribution; the sum over α is the sum over all systematic shifts and σ_α are the magnitudes of δ_α that correspond to the 1σ confidence that δ_α is null.

To obtain the estimates of the neutrino oscillation parameters, Equation 6.6 is minimized over $\vec{\theta}$ and $\vec{\delta}$. Similar to the case without systematic uncertainties, confidence intervals on the oscillation parameters can be obtained in the Gaussian limit. For instance, Figure 6.6 shows an example of the NuMu disappearance contours obtained in the Gaussian limit with and without systematic shifts. As evident from Figure 6.6, the addition of the systematic uncertainties widens confidence contours and makes estimates of the oscillation parameters less precise.

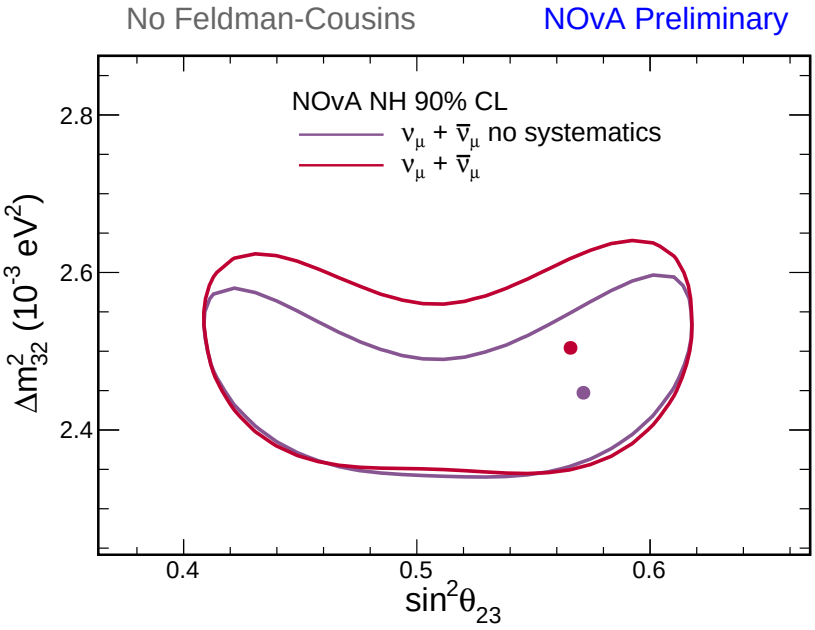


Figure 6.6: Example of sensitivity contours in the Δm^2_{32} vs $\sin^2\theta_{23}$ plane for the NuMu disappearance analysis. The violet dot and contour represent the best fit values of the oscillation parameters and their 90% confidence interval obtained without considering systematic shifts. The magenta dot and contour represent the best fit values of the oscillation parameters and their 90% confidence interval obtained with systematic shifts.

6.6.1 Construction of Predicted Neutrino Spectra as a Function of Systematic Shifts

The procedure to estimate the neutrino oscillation parameters by minimizing Equation 6.6 relies on an implicit assumption that we can construct neutrino energy spectra $N_i^{\text{Pred}}(\vec{\theta}, \vec{\delta})$ as functions of the systematic shifts δ_α. In this subsection, I will touch on how such functions are constructed.

The construction of the neutrino energy spectra as a function of the systematic shifts starts with enumerating all possible systematics of the NOvA experiment and estimating energy spectra $N_{i,\alpha}^{\text{Pred}}(\vec{\theta}, n\sigma_\alpha)$ that correspond to the systematic shift parameter δ_α varied by $n\sigma_\alpha$ for a fixed set of integers n (usually $n \in \{0, \pm 1\}$ or $n \in \{0, \pm 1 \pm 2\}$).

For example, in the case of systematic on the absolute value of the calorimetric energy (Calibration systematic), we expect that the actual value of the calorimetric energy lies within 5% of the predicted energy with confidence 1σ. In order to model this systematic, we create three sets of the MC files corresponding to different values of the systematic shift. The first one is the nominal MC. The second one is an MC where calorimetric energies are shifted by +5% up in order to model $+1\sigma$ systematic shift. And the third one is an MC where calorimetric energies are shifted by -5% down in order to model -1σ shift. Three separate neutrino predictions are constructed on this set of MCs $N_{i,\text{Calibration}}^{\text{Pred}}(\vec{\theta}, n\sigma_{\text{Calibration}})$ for $n \in \{-1, 0, +1\}$.

After a set of predicted neutrino energy spectra $N_{i,\alpha}^{\text{Pred}}(\vec{\theta}, n\sigma_\alpha)$ is constructed for a given systematic α, it is used to create a continuous function $N_{i,\alpha}^{\text{Pred}}(\vec{\theta}, \delta_\alpha)$ of a δ_α parameter. Such a function is constructed by fitting a cubic spline through a set of points $N_{i,\alpha}^{\text{Pred}}(\vec{\theta}, n\sigma_\alpha)$ for each bin i.

6.7 Concluding Remarks

In the previous three chapters, I have described the standard NOvA experiment simulation, reconstruction, and analysis chains that are used to make measurements of the neutrino oscillation parameters. In the next chapter, I will discuss new reconstruction algorithms that I have developed to improve the performance of the NOvA

experiment. After that, I will show the degree of possible improvement of the NOvA performance that can be achieved by the application of these new algorithms.

Chapter 7

Improving Analysis with Recurrent Neural Networks

In this section, I will describe my work on using Recurrent Neural Networks (RNN) in order to improve reconstruction at the NOvA experiment. I will begin this section with an overview of neural networks and their applications. Then, I will expand on my work on developing a novel neutrino energy estimator based on an RNN. Finally, I will discuss the development of a new neutrino event classifier also based on a Recurrent Neural Network.

7.1 Recurrent Neural Networks

7.1.1 Brief History of Artificial Neural Networks

Since the discovery of neurons in animal brains and understanding of their mechanisms of working, computer scientists were curious to reproduce the human brain using a set of artificial neurons. In order to model a biological neuron, an approximation was chosen where an artificial neuron was defined as a function of multiple scalar inputs $\{x_i\}$ (modeled after biological synapses) with a single output (modeled after axon) [52].

A single artificial neuron aggregates information from its inputs using a linear combination $y = \sum_i w_i x_i$, where the scalar values w_i are known as the weights of

the neuron. Then, the artificial neuron applies some nonlinear function f (called activation) over this linear combination y and returns the resulting value $z = f(y) = f(\sum_i w_i x_i)$ as its output [53] (Figure 7.2). The weights w_i of a neuron and its activation function f are the parameters that define a given neuron. In the artificial neural network, the activation function f is usually fixed by an experimenter, and Machine Learning procedures are used to find the neuron weights w_i (in a phase known as training).

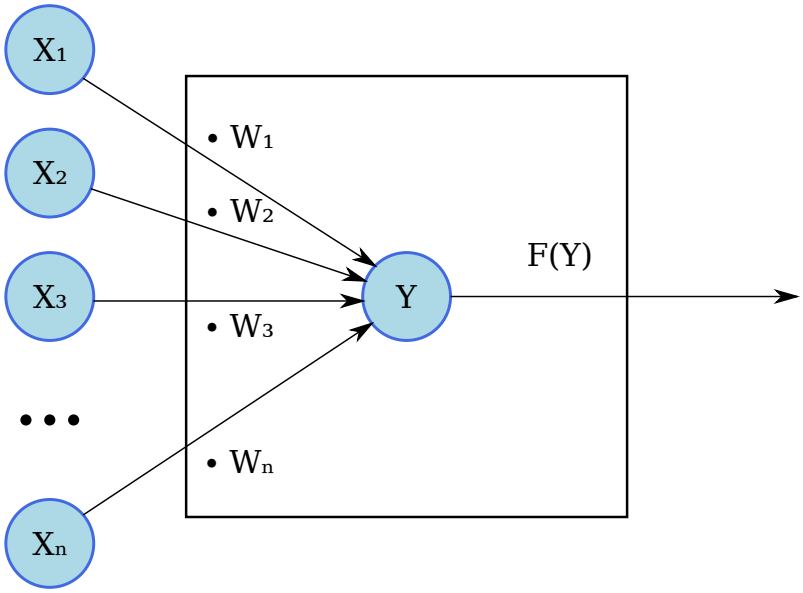


Figure 7.1: Schematic representation of an artificial neuron Y. The blue circles X_i on the left represent inputs of the neuron Y. The values W_i are the weights of the neuron and the function F is an activation function of the neuron. The artificial neuron calculates a linear combination of its inputs $Y = \sum_i W_i X_i$ and returns $F(Y)$ as its output.

In order to model an animal brain, researches tried to arrange neurons into layers, with information flowing sequentially from one layer to the next – a neural network architecture known as a multilayer perceptron [53] (e.g. Figure 7.2). Outputs of neurons in one layer of such a network are used as inputs of the next layer. While the initial experiments with the artificial neural networks were very promising, it was quickly discovered that the neural networks need to be sufficiently large to be able to model complex functions. The large neural networks, however, required significant

computational resources which limited the adoption of neural networks.

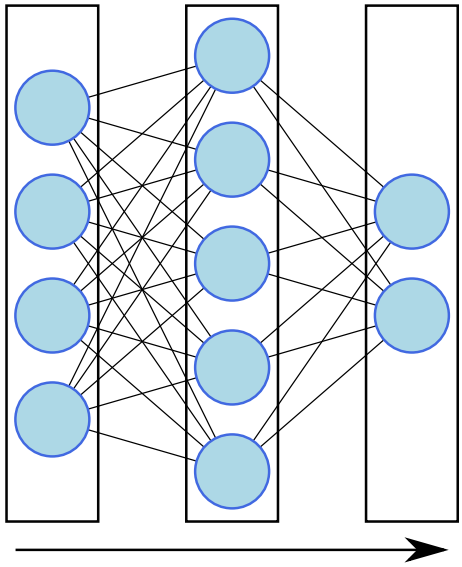


Figure 7.2: Schematic representation of an artificial neural network with three fully connected layers. Each blue circle represents a single neuron. The edges represent connections between neurons. Information flows through these connections from the neurons on the left to the neurons on the right.

The situation changed around 2009-2012 with the development of General Purpose Graphics Processing Units (GPGPU). The early versions of GPUs were optimized for graphics processing for video games and animations. But with the advancements of technology, it became possible to run general-purpose calculations on such GPUs. Unlike a CPU that has a few highly performant and powerful cores, GPUs are made of multiple (thousands) of slow cores, suitable for a massively parallelized evaluation of simple tasks. The GPU cores are the perfect fit for the evaluation of neural networks since the neural network evaluation requires a parallel computation of a simple function over many neurons.

In 2012, a large neural network (known as AlexNet) was trained on two GPUs [55] for the purpose of image classification and it vastly outperformed all traditional image classification methods. This opened a new era of neural network exploration where previously computationally intractable neural networks were revisited and new more powerful networks were constructed [54].

7.1.2 Recurrent Neural Networks

The simple layered neural networks discussed in the previous paragraph are not suitable for all tasks. One of the tasks for which the traditional neural networks are not suitable is the handling of sequential data. For example, the task of predicting weather today (or stock market movement, or next word in a sentence) based on observations of weather on previous days. An algorithm capable of such a task must be able to collect weather observations from multiple previous days (with possibly varying number of days) and combine them efficiently into a prediction about today's weather. While the simple neural networks can be extended to use weather observations of N previous days as inputs and predict the weather today, they will be highly inefficient, since they won't be able to exploit the fact that weather inputs from day N-1 and day N-2 have a very similar structure (they are homogeneous). They also won't be able to handle a situation when N varies.

In order to handle sequential data new classes of algorithms were explored. One class of such algorithms is known as latent autoregressors. These algorithms work by encoding previous observations into some state (sometimes called memory). Then, they use this memory in order to arrive at the desired prediction.

The simple feed-forward neural network from the previous section can be used in order to construct a latent autoregressor algorithm (e.g. Figure 7.3). Such a neural network reads its inputs sequentially. When it processes a single input it combines its information with the memory of past inputs and outputs an updated memory. Finally, another simple neural layer can be used to predict the required quantity from the final memory state. A class of neural networks that work in this way is known as Recurrent Neural Networks.

Unfortunately, experimental observations show [56] that simple networks like Figure 7.3 tend to forget past information very quickly and are very difficult to train due to the accumulation of numeric errors when processing long sequences. To address the issue of accumulation of numerical error an alternative architectures of Recurrent Neural Networks were developed. One of such architectures is the Long Short-Term Memory Network (or simply LSTM) [56]. This type of neural network uses special "gates" (inspired by the logical gates in computer RAM) that control whether to remember past information or forget it. The explicit forgetting

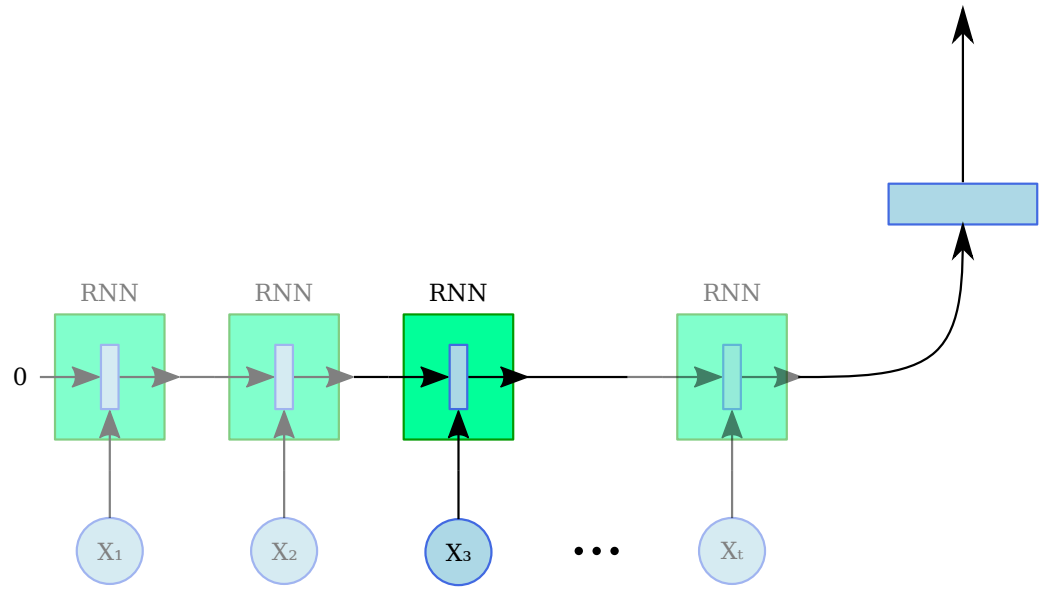


Figure 7.3: Schematic representation of a simple Recurrent Neural Network (green block). A simple RNN is made of a single layer of neurons. It processes inputs X_t sequentially creating a state that encodes a memory of previous inputs. Finally, another neural layer (blue block) is used to predict the desired quantity from the RNN memory.

of past information stops numerical errors from the past to influence the future and substantially improve the network performance.

The LSTM neural network is the basis of the neutrino energy estimator and event classifiers that I have designed for the NOvA experiment and I will discuss their development in the following sections.

7.2 LSTM Energy Estimator

7.2.1 Energy Estimation for the NuMu Disappearance Analysis

As was discussed previously, the NuMu Disappearance analysis at NOvA is focused on the selection and reconstruction of the Charged Current ν_μ interactions. The NuMu CC interaction has a distinctive event topology shown in Figure 7.4 – typically

in these events one can observe a long clear muon track and a hadronic activity around the interaction vertex. Correspondingly, the NOvA experiment developed its first ν_μ-CC energy estimation algorithm taking advantage of this topology.

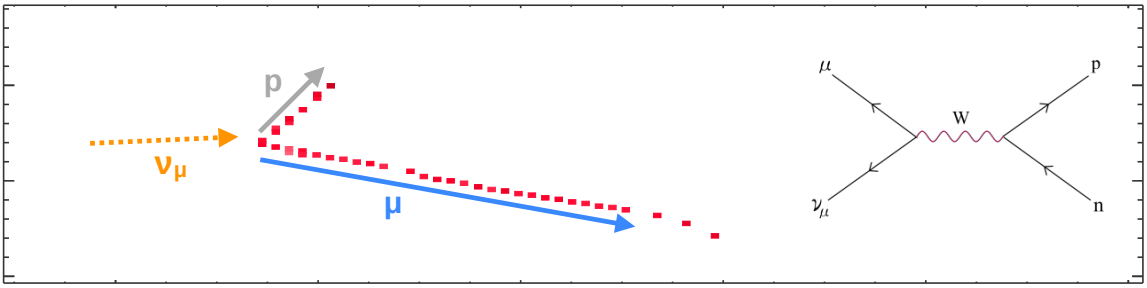


Figure 7.4: Event topology of the NuMu CC interactions

I have discussed the standard NOvA NuMu CC ($\nu_\mu \to \mu$+Had) energy estimation algorithm (also known as the Spline EE) in subsection 5.5.1. Below is a summary of how this energy estimator works:

1. First, it finds a track of a muon and predicts muon energy E_μ from the length of the track (using a piecewise linear spline as a model).

2. Second, it collects all hits that do not belong to the muon track and finds their total calorimetric energy. Then, it tries to predict the hadronic energy component E_{had} from the total calorimetric energy of non-muon hits (using another piecewise linear spline).

3. Finally, by the law of energy conservation, the energy of the incoming neutrino is simply a sum of the muon and hadronic energy parts: $E_{\nu_\mu} = E_\mu + E_{\text{had}}$.

The Spline Energy Estimator has a relatively simple structure, yet it achieves a reasonably good energy reconstruction. That is why it has been used at the NOvA experiment for four rounds of analysis. However, it also does show its age mostly because it relies on two very coarse objects to estimate the ν_μ energy – the muon track length and the total calorimetric energy. That was appropriate for the early NOvA analysis, but since that time many advanced algorithms have been developed that operate on the individual prongs. These algorithms are capable of finding particle

tracks in each prong, predicting the type of the particle that produced a given prong, estimating geometric and kinematic properties of each prong, etc.

The Spline Energy Estimator is not able to use any additional relevant information provided by such algorithms, and hence its predicted energy is suboptimal. Therefore, we have tried to develop a new algorithm, that is able to incorporate many new variables from multiple advanced reconstruction algorithms. In particular, we tried to develop an algorithm that can process information from the individual prongs.

Working with prong level variables proved out to be challenging since the number of prongs varies in each event. Therefore, we were looking for an algorithm that can handle data with a variable number of inputs. The recurrent neural networks turned out to be the natural candidate for such an algorithm since they are able to process inputs in a sequential manner.

7.2.2 The LSTM Energy Estimator Concept

Originally, an energy estimator that is capable of working with the prong level variables was proposed by Alexander Radovic [57]. That Proof of Concept energy estimator used an LSTM neural network unit to process prong level variables and then utilized simple neurons to combine the output of the LSTM network with the slice level calorimetric energy in order to estimate the total ν_μ energy of the event and the muon energy component (Figure 7.5).

The original energy estimator relied only on a few input variables, summarized in Table 7.1. The prong level inputs include:

1. The energy, momenta, and overlap energies predicted by the Break Point Fitter algorithm.
2. The calorimetric energy of the prong.
3. The prong particle ID predicted by the Prong CVN classification algorithm.
4. The direction and length of the prong.

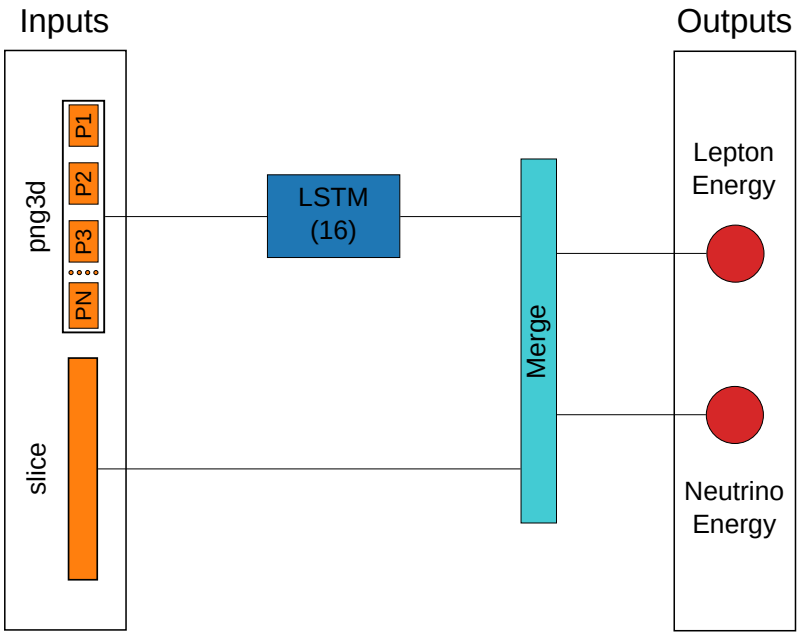


Figure 7.5: Architecture of the LSTM based neural network used in the Proof of Concept recurrent neural network energy estimator.

The slice level inputs contain only the calorimetric energy of the slice as a whole. Despite having a very simple network architecture, the LSTM based energy estimator had better performance than the original spline-based energy estimator (Figure 7.6).

7.2.3 LSTM EE Development: Network Architecture and Input Variables

Due to the initial success of the Proof of Concept LSTM energy estimator, it was decided to continue its development. The development started with the studies of the effect of model complexity and input variables on the overall performance.

Initial Model Complexity Studies

The original LSTM energy estimator architecture (Figure 7.5) is very simplistic and there is a big potential for further improvements. Multiple avenues of LSTM EE architecture modification have been investigated [58]. The biggest improvements were

3D prong vars:	**Slice vars:**

`png.bpf[i].energy`
`png.bpf[i].momentum.{x,y,z}`
`png.bpf[i].overlapE`
`png.calE`
`png.cvnpart.{`
`  muon,pion,proton,`
`  electron,photonid`
`}`
`png.dir.{x,y,z}`
`png.len`
`png.weightedCalE`
	`calE`
	`remPngCalE(= calE - total png.calE)`

Table 7.1: Summary of the inputs that the original Proof of Concept LSTM energy estimator was using. The left column shows prong level variables and the right column shows the slice level variables.

observed from modifications that add input normalization and from the modifications that add pre and post processing layers to the LSTM unit. Optimizing model architecture I have arrived at the second version of the LSTM energy estimator shown in Figure 7.7.

The new network architecture (v2) employs Batch Normalization layers to perform input normalization and has several Dense layers for input preprocessing and postprocessing of the LSTM network output. It has better performance than the original network (Figure 7.5). For example, in terms of the RMS of the energy resolution, it has 7.5% (Figure 7.8) compared to the RMS of the original LSTM EE (Figure 7.6) of 8.1%.

Input Variable Refinement

To further improve the performance of the energy estimator we have tried to revise the input variables that it receives. The following changes were considered:

- First, the Near Detector has a muon catcher region, with significantly different properties than the rest of the detector. Correspondingly, we cannot apply the same energy estimator in the muon catcher as in the active region of the detector. In the standard NOvA energy estimation algorithm, this issue is

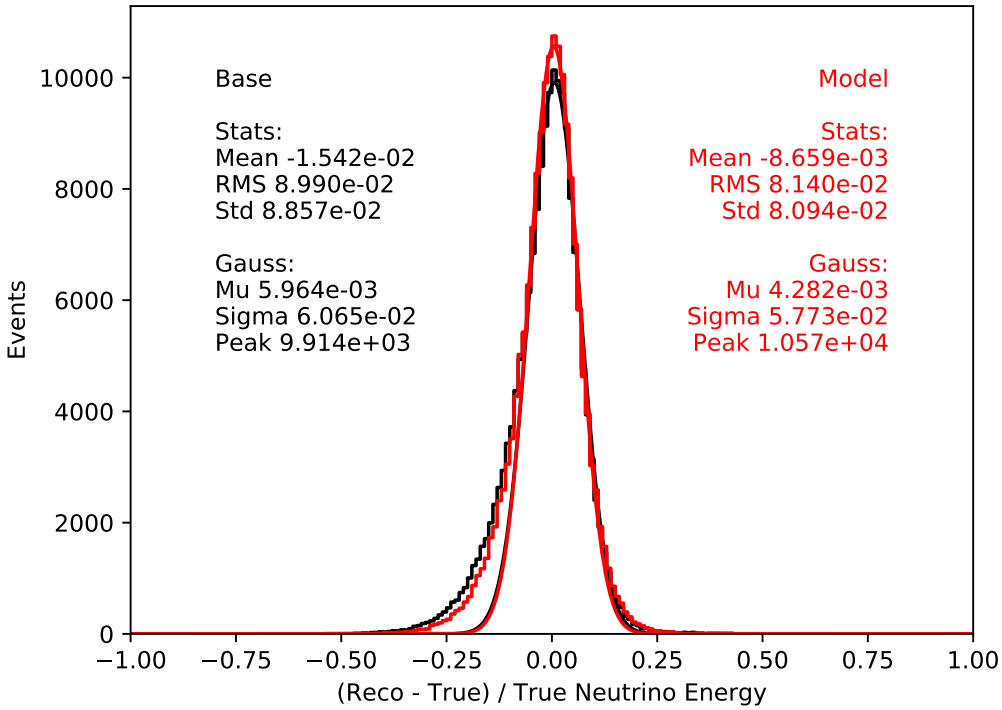


Figure 7.6: Energy resolution histograms for the spline-based energy estimator (black) and the original LSTM energy estimator (red). In terms of the RMS of the energy resolution, the LSTM energy estimator outperforms the spline-based one 8.1% vs 9.0%. The Gaussian fits (smooth curves) were performed for the peaks of both energy resolution histograms. In terms of the spread (sigma) of the Gaussian fits, the LSTM energy estimator also achieves better performance 5.8% vs 6.1%.

solved by having two different spline functions for the ND active region and muon catcher region. However, due to the flexibility of neural networks, it was decided that instead of having different LSTM energy estimators for different parts of the detector, one can instead supply prong starting positions to the energy estimator and let it figure itself whether particle enters muon catcher or not.

- Second, it was discovered previously [59] that for the first two NOvA periods one must use different energy estimation algorithms at the FD. Therefore, the standard NOvA energy estimation algorithm uses different spline fits for the first two NOvA periods and the remaining runs. However, similarly to the

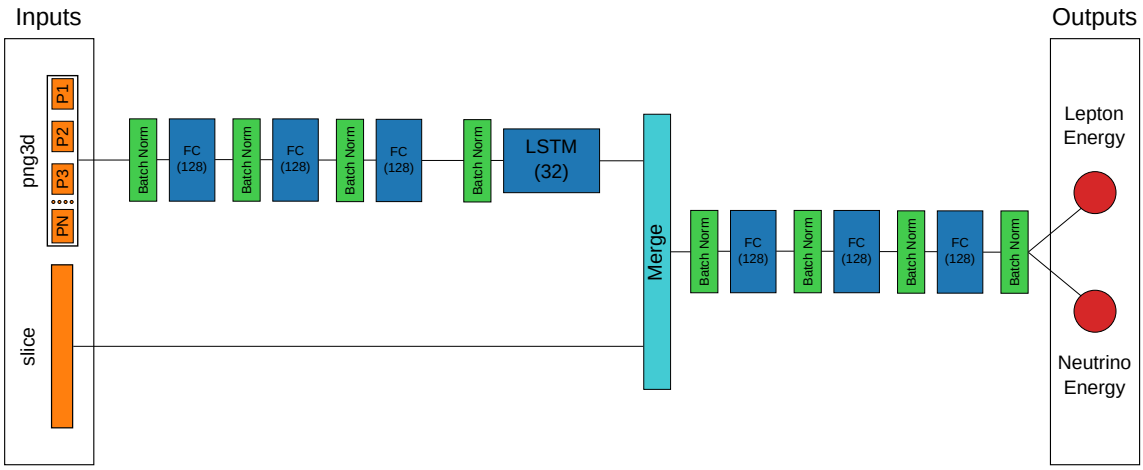


Figure 7.7: Second version of the LSTM energy estimator architecture. The green blocks denote the Batch Normalization layers. The blue rectangular blocks are the fully connected layers with the `ReLU` activation function.

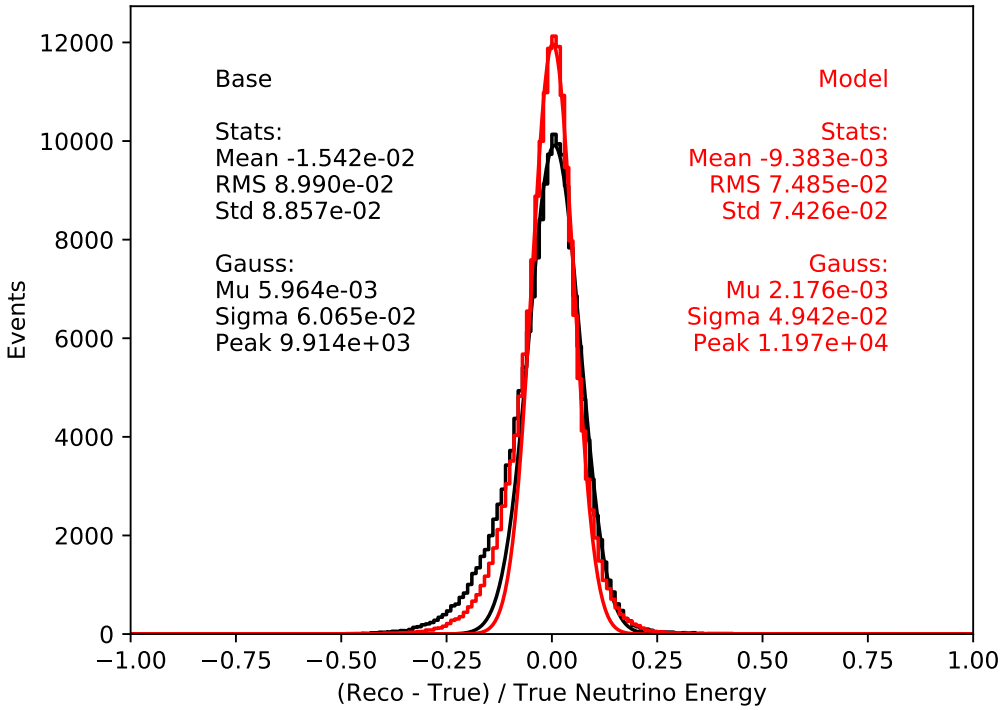


Figure 7.8: Energy resolution histograms for the spline-based energy estimator (black) and the LSTM energy estimator v2 (red).

previous point, it was decided not to train multiple LSTM energy estimators, but instead provide flags to the EE, which will indicate whether it is period 1, period 2, or the remaining periods of the NOvA operation.

- Next, it was judged that the number of hits/planes in prongs might be a useful metric for estimating energies. Therefore, they were also added to inputs.

- Finally, the original set of input variables did not capture all objects with energies in the event. It was missing variables associated the the two-dimensional prongs. These are the prongs that were failed to be matched in X and Y views into the 3-dimensional objects. Since the 2D prongs have a different structure than the 3D prongs, it was decided to add a separate LSTM unit into the network to process them.

The refined set of input variables is summarized in Table 7.2. Compared to the original set of inputs (Table 7.1) I have added 2-dimensional prong inputs, prong starting positions, and the number of hits/planes in prongs. At the slice level, I have added two flags `IsCoarseTiming` and `IsLowGain` that can help the energy estimator to distinguish between periods in the Far Detector, and the `orphCalE` variable that contains calorimetric energy not accounted in the 3D prongs.

3D prong vars:	**2D prong vars:**	**Slice vars:**
`vtx.elastic[0].fuzzyk.png`:	`vtx.elastic[0].fuzzyk.png2d`:	
`png.bpf[i].energy`	`png2d.calE`	`calE`
`png.bpf[i].momentum.{x,y,z}`	`png2d.dir.{x,y,z}`	`IsCoarseTiming`
`png.bpf[i].overlapE`	`png2d.len`	`IsLowGain`
`png.calE`	`png2d.nhit{,.x,.y}`	`nHit`
`png.cvnpart.{`	`png2d.nplane`	`orphCalE`
`  muon,pion,proton,`	`png2d.start.{x,y,z}`	`remPngCalE`
`  electron,photonid`	`png2d.weightedCalE`	
`}`		
`png.dir.{x,y,z}`		
`png.len`		
`png.nhit{,.x,.y}`		
`png.nplane`		
`png.start.{x,y,z}`		
`png.weightedCalE`		

Table 7.2: Summary of the refined inputs of the LSTM energy estimator.

As I stated above, the incorporation of the 2D prong inputs required modifying the network architecture and adding a separate LSTM unit branch to handle them. The updated network architecture is shown in Figure 7.9. Training the LSTM energy estimator with these refinements [58] shows further improvements in the energy resolution (Figure 7.10). Compared to the second version of the LSTM energy estimator architecture (Figure 7.10) we see an improvement of the RMS value from 7.5% to 6.8%. Or comparing the performance of the refined LSTM energy estimator to the performance of the NOvA standard spline-based EE we see even more drastic improvement of the RMS from 9.0% to 6.8% (or 24% in relative terms).

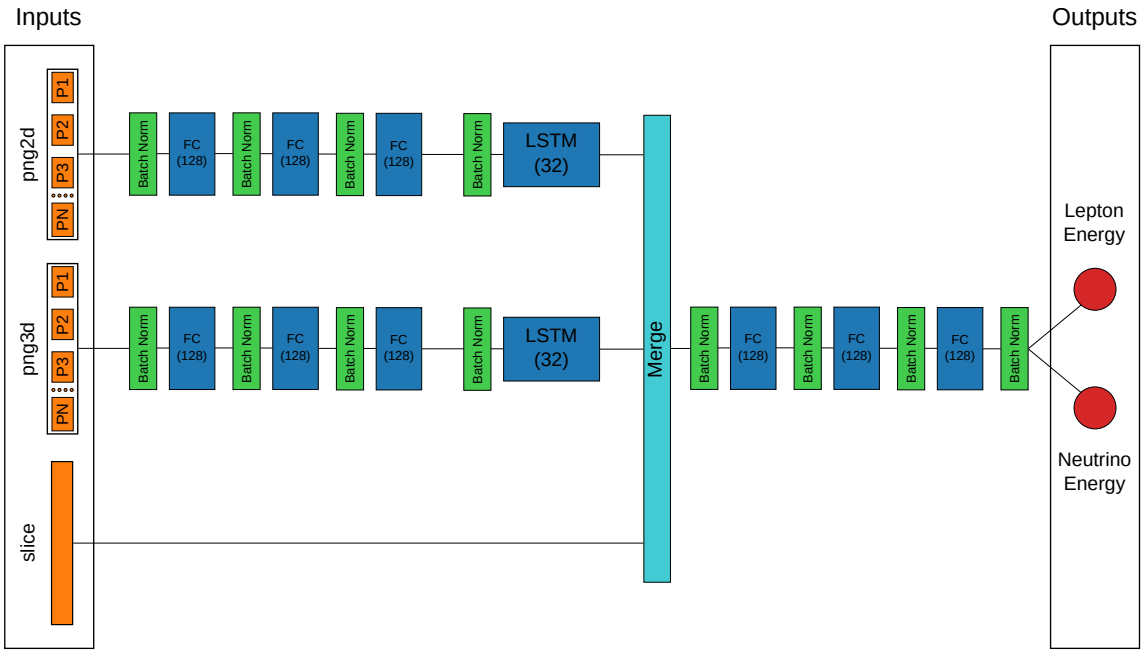


Figure 7.9: Third version of the LSTM energy estimator architecture. The green blocks denote the Batch Normalization layers. The blue rectangular blocks are the fully connected layers with the `ReLU` activation function.

7.2.4 LSTM EE Development: Sample Selection

The NOvA NuMu Disappearance analysis uses events with high PID scores (i.e. closely resembling the typical ν_μ interaction shown in Figure 7.4) and with energies up to 5 GeV. However, the energy estimator might be also used for other analysis types

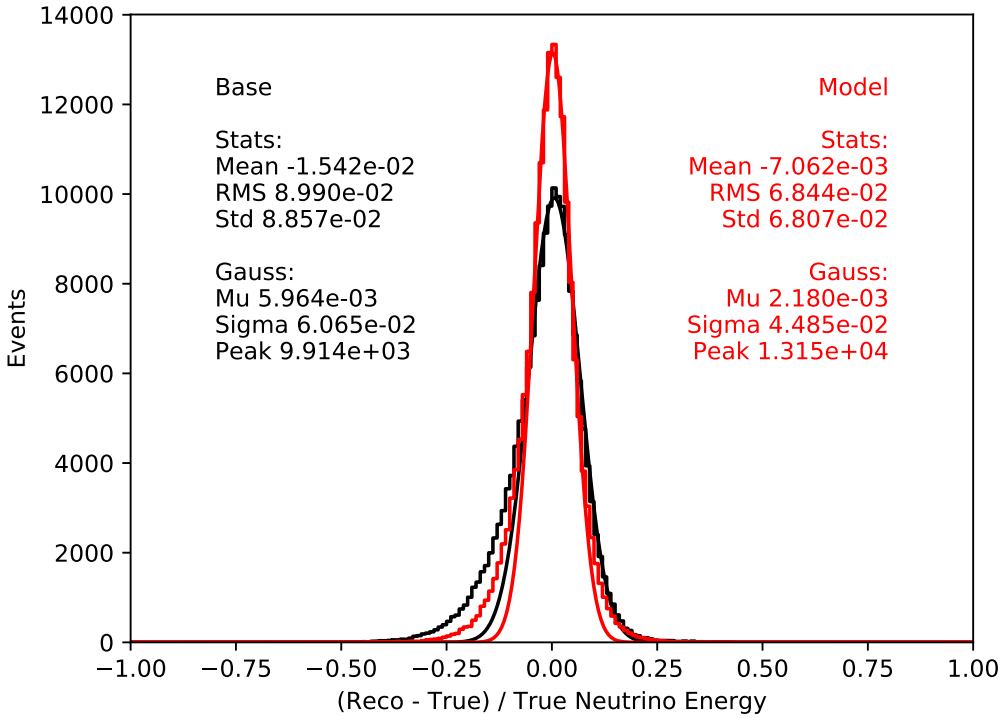
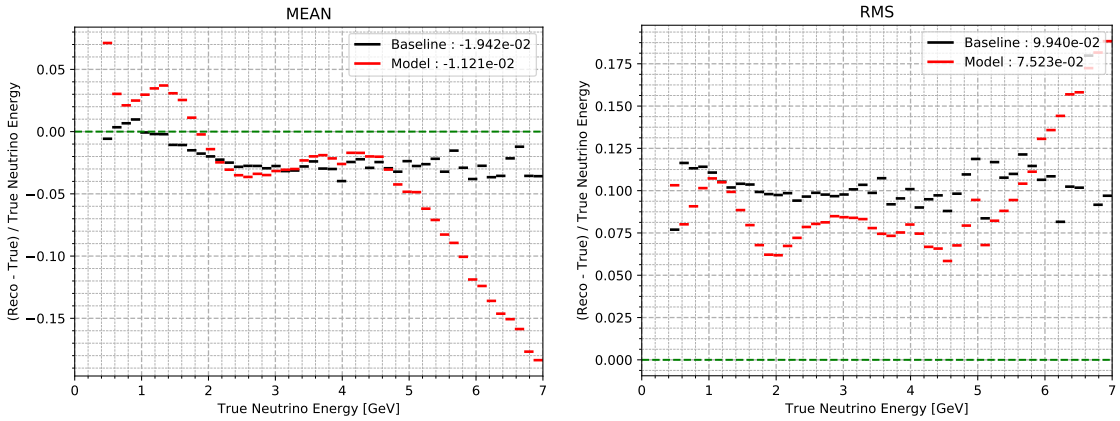


Figure 7.10: Energy resolution histograms for the spline-based energy estimator (black) and the LSTM energy estimator v3 (red).

with more loose requirements on the event PID score and energy. Correspondingly, we have to make sure that our energy estimator performs reasonably well for such events. A study was conducted to see how the energy estimator trained on the sample with the standard NuMu Disappearance preselection (will be referred to as the Standard Sample) performs on a sample with a looser preselection (will be referred to as the Loose Sample) [60].

To better understand such a performance, plots of the energy resolution vs true neutrino energy bins were made (Figure 7.11). From the left side of Figure 7.11 (means of the energy resolution), one can see that the energy estimator trained on the Standard Sample has a high bias at energies above 5 GeV (the red curve goes down from about 2% bias at 4 GeV to about 20% bias at 7 GeV) when evaluated on the Loose Sample. Similarly, the plot of the RMS of the neutrino energy resolution, presented on the right side of Figure 7.11, shows that the RMS of the LSTM energy

estimator quickly degrades above 5 GeV, becoming even worse than the RMS of the standard spline-based energy estimator. Such behavior was judged too bad and I tried to correct it.



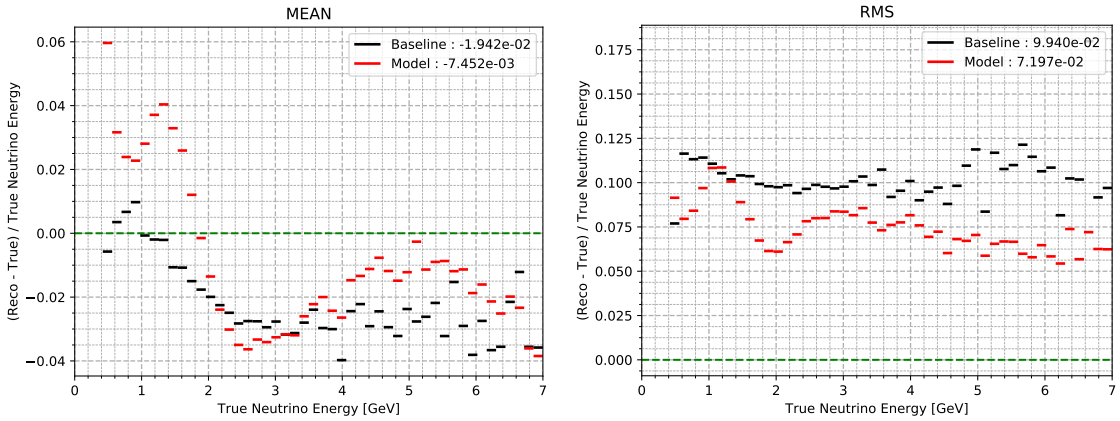
(a) Means of Energy Resolution

(b) RMS of Energy Resolution

Figure 7.11: Plots showing statistics of the ν_μ energy resolution (Reco - True) / True plotted vs true neutrino energy bins. The left plot shows the means of the ν_μ energy resolution, the right one shows the RMS of the ν_μ energy resolution. The red distribution is for the LSTM energy estimator (trained on the Standard Sample), while the black one is for the spline-based energy estimator. Both plots are made on the Loose Sample.

This behavior is the manifestation of the well-known problem in machine learning – the problem of extrapolation. The LSTM energy estimator was trained on the Standard Sample, but now it is evaluated on the Loose Sample, which contains events (in particular events with energies above 5 GeV) that it did not see during the training. Since the LSTM EE has never seen such events, it cannot predict their energy properly. Therefore, to improve the performance of the LSTM energy estimator on the Loose Sample one has to train it on the Loose Sample itself.

I have retrained the LSTM energy estimator on the Loose Sample. Figure 7.12 shows performance plots for the LSTM EE trained on the Loose Sample, similar to the plots shown in Figure 7.11 for the EE trained on the Standard Sample. Comparing Figure 7.11 and Figure 7.12 we can observe that training LSTM EE on the Loose Sample does indeed solve poor energy estimation at energies above 5 GeV.



(a) Means of Energy Resolution

(b) RMS of Energy Resolution

Figure 7.12: Plots showing statistics of the ν_μ energy resolution (Reco - True) / True plotted vs true neutrino energy bins. The left plot shows the means of the ν_μ energy resolution, the right plot shows the RMS of the ν_μ energy resolution. The red distribution is for the LSTM energy estimator(trained on the Loose Sample), while the black one is for the spline-based energy estimator. Both plots are made on the Loose Sample.

However, the retraining on the Loose Sample might have degraded the performance of the LSTM energy estimator on the Standard Sample that we are primarily concerned about. To verify whether it is the case, I have evaluated the performance of the LSTM EE trained on the Loose Sample on the original Standard Sample (Figure 7.13). Comparing the performance of the LSTM EE trained on the Loose Sample (Figure 7.13) to the performance of the LSTM EE trained on the Standard Sample (Figure 7.10) we actually observe a slight improvement in the overall energy resolution (both in terms of the RMS and Gaussian spread) and no degradation of performance. I believe this improvement comes from a fact that the Loose Sample has much larger statistics than the Standard Sample, which is often beneficial for training.

Due to these observations, it was decided that the training on the Loose Sample is better than the training on the Standard Sample. Therefore, from this point, all the networks will be trained on the Loose Sample, unless explicitly specified otherwise.

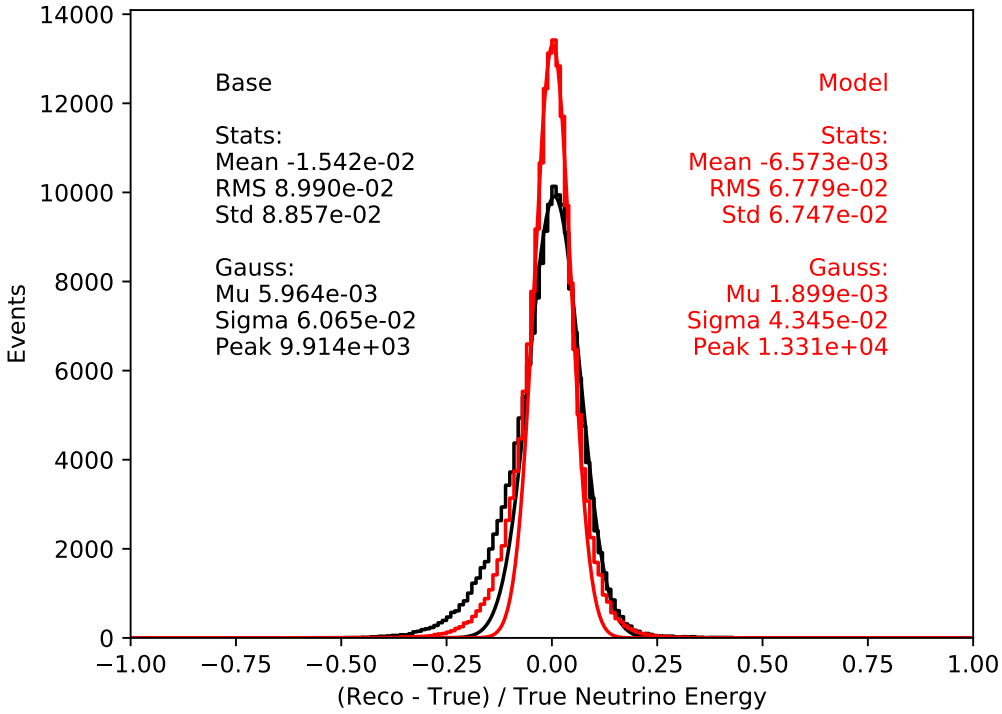


Figure 7.13: Energy resolution histograms for the spline-based energy estimator (black) and the LSTM energy estimator trained on the Loose Sample (red). The plot is made on the Standard Sample.

7.2.5 LSTM EE Development: Low Energy Bias Correction

While performing detailed studies of the performance of the LSTM energy estimator another issue was discovered. When we look at the plot of energy resolution vs true energy for the Standard Sample (Figure 7.14), we see that the LSTM EE has a much larger bias in the range of the true energy between $1-2\,\text{GeV}$. Since the range between $1-2\,\text{GeV}$ is the range where we expect to observe the neutrino oscillation, it is imperative to have as unbiased an estimator in that region as possible.

The bias at low energies was traced to the composition of the training sample. Figure 7.15 shows the true neutrino energy distribution in the Standard Sample. The true energy distribution has a peak around $2\,\text{GeV}$. Therefore, the LSTM energy estimator can achieve good performance by accurately estimating energies around the peak, while having a subpar performance outside of the peak.

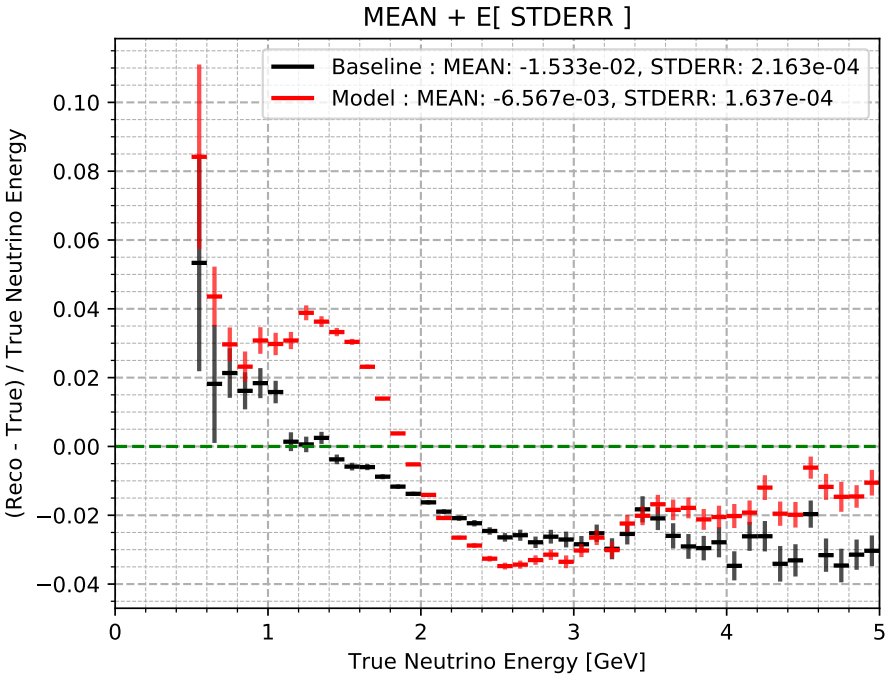


Figure 7.14: Means of the neutrino energy resolution (Reco - True) / True plotted vs true neutrino energy bins. The means of the LSTM energy are shown in red, and the means of energies predicted by the Spline EE are shown in black. The LSTM energy curve (red) has a clear bias between $1 - 2\,\text{GeV}$.

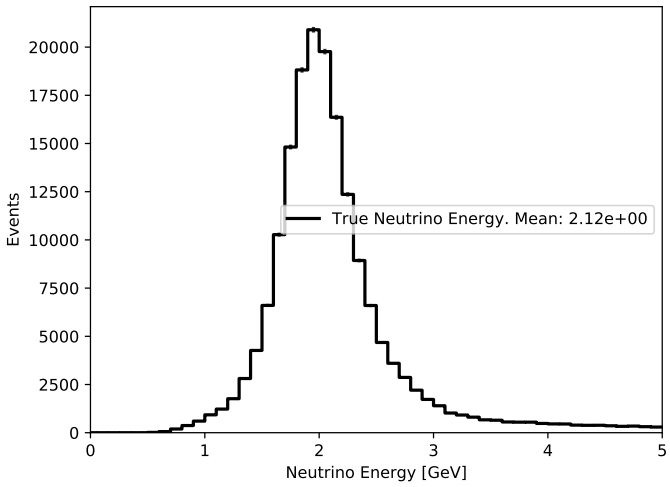
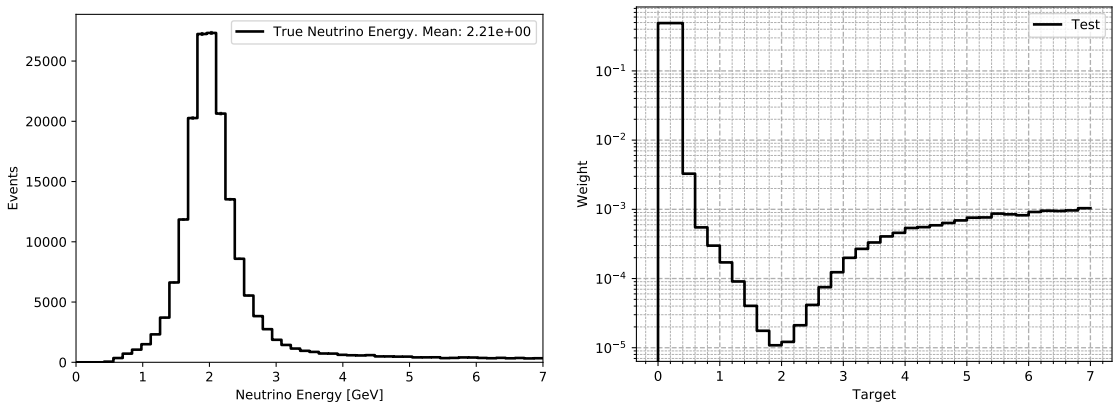


Figure 7.15: True neutrino energy distribution in the Standard Sample.

Consequently, to lower the bias in the region between $1-2\,\mathrm{GeV}$ we needed to make the training sample have a flat energy distribution between $1-2\,\mathrm{GeV}$. We have considered several options to achieve this. First, since the training sample was generated with the peak around $2\,\mathrm{GeV}$, it was suggested to regenerate it with the flat energy distribution from the beginning. However, this turned out to be a complex task since it required changing the underlying simulation model. Therefore, we have resorted to machine learning methods to correct the peaked distribution. The data reweighting method was judged to be the simplest one to implement.

The special weights were constructed to make the training sample have a flat energy distribution. To construct these weights we have started with the original true energy distribution in the sample. Then, we took the inverse of that distribution. Figure 7.16 depicts these steps. The weight for each event is determined by first finding the true energy bin number where this event falls into. And then using the value from the corresponding weight bin in Figure 7.16 (with a proper normalization) as the event weight.



(a) True Energy Distribution (b) Weights for True Energy Bins

Figure 7.16: Illustration of the Flat Weights construction for the Loose Sample. The left pane shows the original true energy distribution in the sample. The right pane shows the inverse of that energy distribution (log scale).

I have retrained the LSTM energy estimator with the above-defined Flat Weights [61]. Retraining with the Flat Weights does indeed improve bias in the low energy region $1-2\,\mathrm{GeV}$. This can be seen comparing the old energy resolution plot vs true neu-

trino energy (Figure 7.14) to the corresponding plot for the LSTM EE trained with the Flat Weights (Figure 7.17).

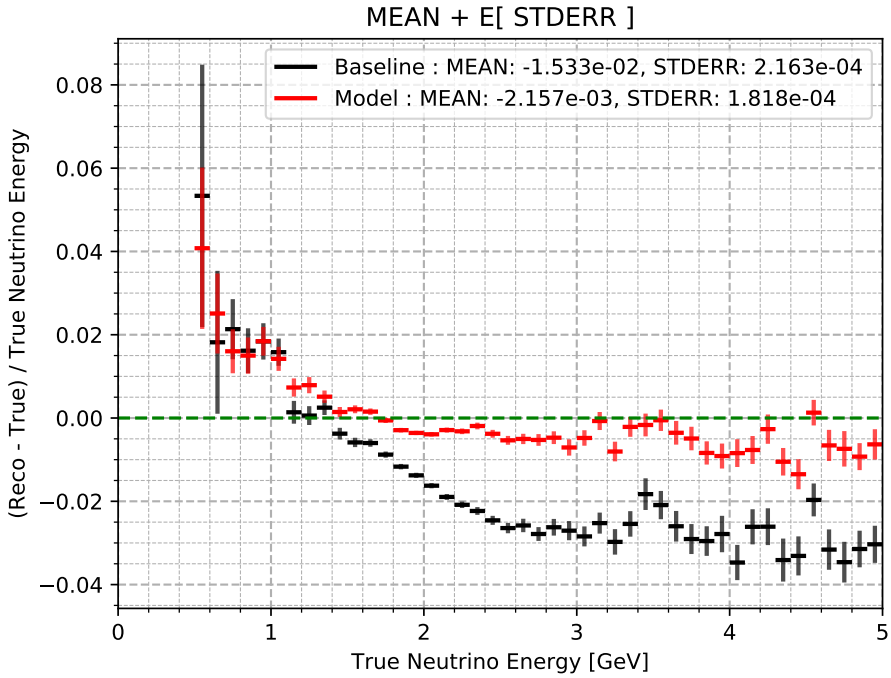


Figure 7.17: Means of the neutrino energy resolution (Reco - True) / True plotted vs true neutrino energy bins. The means of the LSTM energy are shown in red, and the means of energies predicted by the Spline EE are shown in black. The LSTM EE is trained with the Flat Weights.

Unfortunately, the bias reduction for low energy events was achieved at expense of lowering the overall energy resolution of the LSTM energy estimator. For example, looking at the energy resolution plot for the LSTM EE with the Flat Weights (Figure 7.18) and comparing it to the LSTM EE trained without the Flat Weights (Figure 7.13) we see that the RMS degrades from 6.8% down to 7.5% with the addition of the Flat Weights.

This can also be observed by comparing plots of the RMS of the neutrino energy resolution per true neutrino energy bins for LSTM EE with and without the Flat Weights (Figure 7.19). From the left plot of Figure 7.19 one can observe that the LSTM EE trained without the Flat Weights is mostly focused on reconstructing neutrino energies around 2 GeV correctly while ignoring the tails of the distribution.

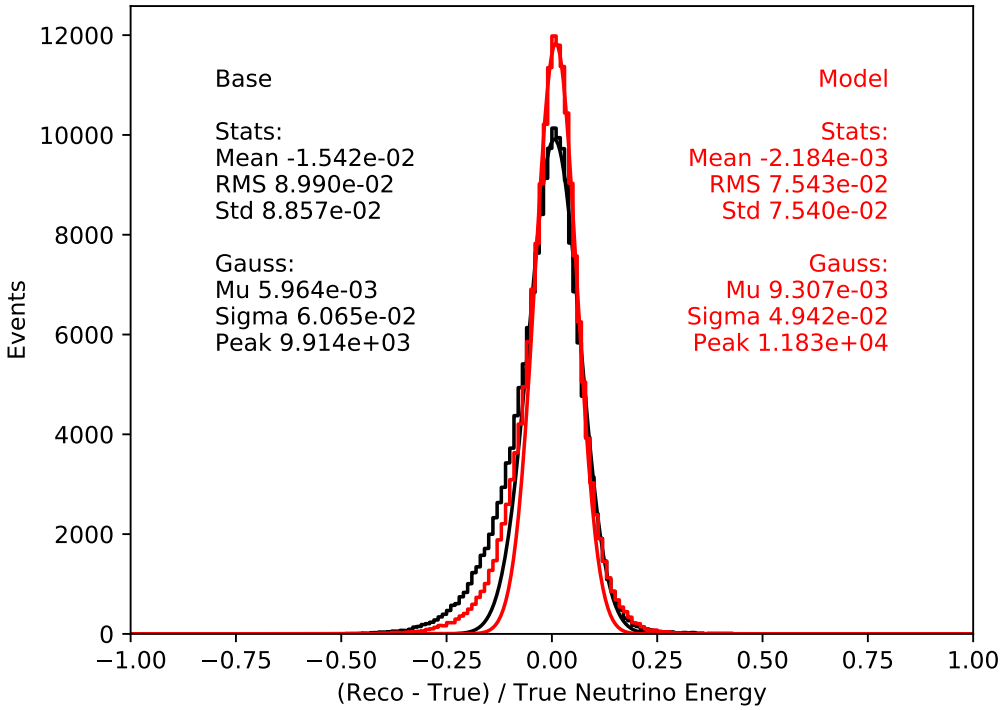


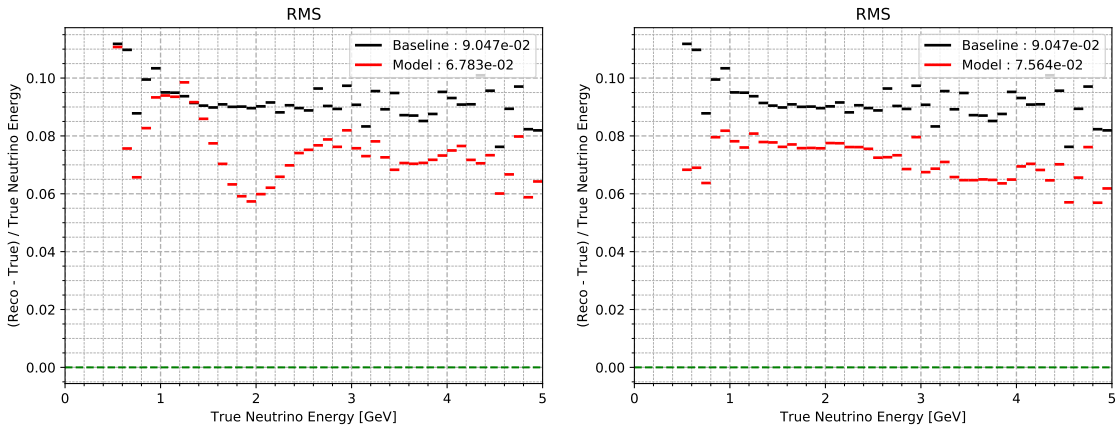
Figure 7.18: Energy resolution histograms for the spline-based energy estimator (black) and the LSTM energy estimator trained on the Loose Sample with the Flat Weights (red). The plot is made on the Standard Sample.

On the other hand, the right plot of Figure 7.19 shows that the addition of the Flat Weights makes the LSTM EE focus on all ranges of energy equally.

It was decided to use the Flat Weights since they give a less biased energy estimator in the oscillation region.

7.2.6 LSTM EE Development: Weights Tuning for the Near Detector

So far we have looked at the training and performance of the LSTM EE at the Far Detector. I have also trained the LSTM energy estimator for the Near Detector, where it shows similar performance to the Far Detector energy estimator [62]. However, the difference in performance between the FD and ND energy estimators appeared when I tried to implement the new features – training on the Loose Sample



(a) No Flat Weights

(b) With Flat Weights

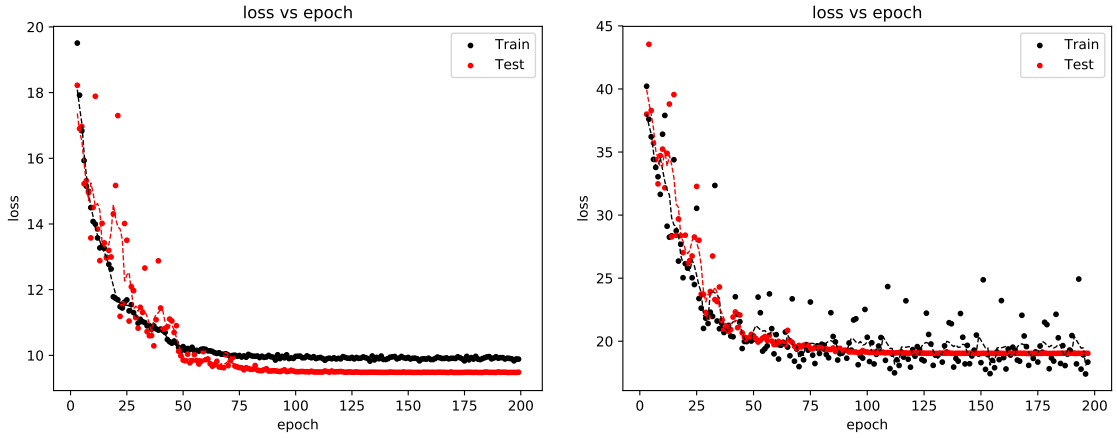
Figure 7.19: Comparison of the RMS of the neutrino energy resolution plotted per true energy bins between the LSTM EE trained on the sample without Flat Weights (left) and with the Flat Weights (right). The red curves show the RMS for the LSTM EE and the black curves for the Spline EE.

and Flat Weights at the Near Detector.

When the ND LSTM energy estimator is trained on the Loose Sample with the Flat Weights, the training procedure itself becomes highly unstable. I have illustrated that in Figure 7.20. Comparing the left pane of Figure 7.20 to the right one, we can see that adding the Flat Weights makes network loss function oscillate significantly.

The result is that the LSTM energy estimator training fails to converge and the energy estimator itself has a rather poor performance overall. For example, the Near Detector LSTM EE trained without the Flat Weights has performance in terms of the RMS of 9.0% (c.f. Figure 7.21), while the LSTM EE trained with the Flat Weights has much worse RMS of 10.7% (Figure 7.22).

I was able to track this degradation of performance to the flaw in the weights construction procedure itself. From the description of the Flat Weights in subsection 7.2.5 – the Flat Weights are obtained by making a true neutrino energy distribution and taking its inverse. This approach worked fine at the Far Detector, but at the Near Detector, there is too much variance between the number of events in the peak of distribution and in the tails. This results in the tails of the Near Detector



(a) No Flat Weights (b) With Flat Weights

Figure 7.20: Training/Validation Loss vs Epoch plots for the LSTM EE training at the Near Detector without the Flat Weights (left) and with the Flat Weights (right). The loss function exhibits high oscillations for the training with the Flat Weights.

energy distribution getting weights about 10 times higher than the tails in the Far Detector distribution (Figure 7.23).

However, events in the tails of the ND distribution are also 10 times as rare as the events at the tails of the FD distribution, so they are rarely seen during the training. During the training, most of the time the network sees events around the peak of the energy distribution and so it learns to estimate their energy. But sometimes an event from the tail of the distribution gets sampled with an enormous weight. That rare event with the huge weight creates a large gradient and moves the network away from the stable configuration. Samplings of such rare events can be seen as multiple bumps in the right pane of the training history plot shown in Figure 7.20.

One way to address this problem is to sample events according to the Flat Weights, instead of using a simple random sampling during the network training. However, this will complicate the training procedure. Another complication is that the tails of the energy distribution simply do not have enough statistics for the proper training, which can be seen comparing the Flat Weights between the training and test samples (Figure 7.24) – the Flat Weights show a high discrepancy between the training and test samples at the tails of the distributions.

Another way to address the problem would be to generate a larger training sample

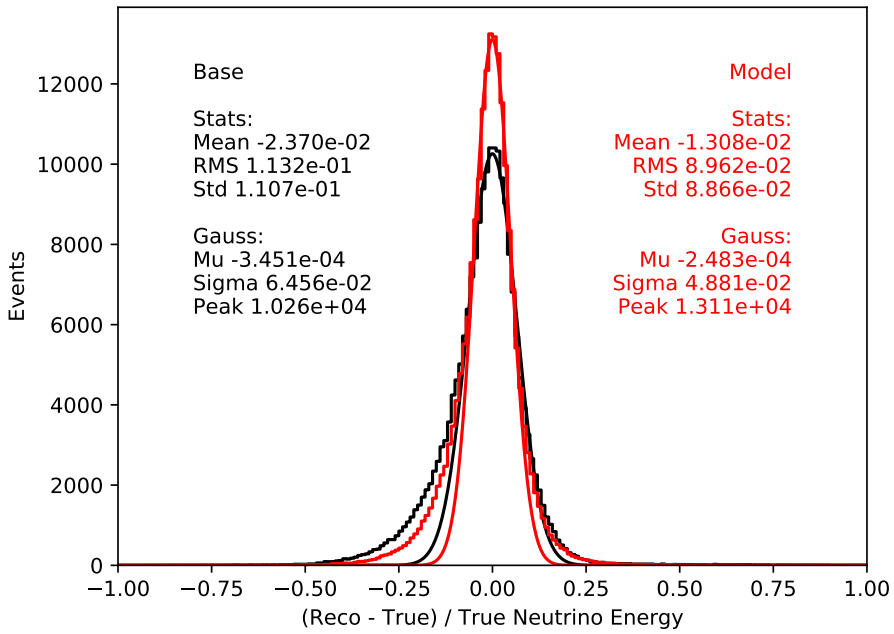


Figure 7.21: Near Detector energy resolution histograms for the spline-based energy estimator (black) and the LSTM energy estimator trained on the Loose Sample without the Flat Weights (red). The energy estimators are evaluated on the Standard Sample.

with enough statistics across all energies. However, that was not feasible due to the production constraints. Therefore, we have solved this problem by clipping the maximum value of the weight that it can achieve. I have trained an ensemble of neural networks with different values of the Flat Weight clip value and different true neutrino energy binnings to find which configuration gives the optimal performance [63].

The optimal performance of the ND LSTM energy estimator is achieved with a finite Flat Weight clipping value around 50 [63] (i.e. the ratio between the maximum value of the weight and the minimum value is bounded by 50). With this weight configuration, the Near Detector LSTM EE shows much better energy reconstruction (Figure 7.25) with the RMS of 9.9% compared to the RMS without clipping of 10.7%. Clipping the Flat Weights also improves training convergence and removes the discrepancy between the weights for the training and test samples [63].

The only downside of using the Flat Weight clipping is the degradation of perfor-

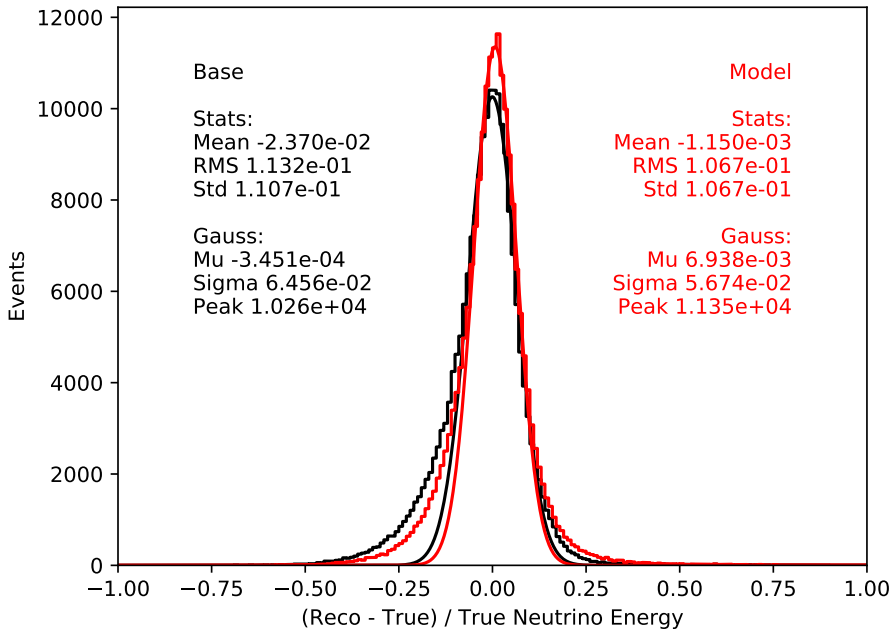


Figure 7.22: Near Detector energy resolution histograms for the spline-based energy estimator (black) and the LSTM energy estimator trained on the Loose Sample with the Flat Weights (red). The energy estimators are evaluated on the Standard Sample.

mance at the high energy tail of the ν_μ energy distribution, which can be seen from Figure 7.26 – means of the ν_μ energy resolution exhibit a high bias at energies above 5 GeV when trained with weight clipping compared to the training without clipping. Overall, the improvement of the energy resolution in the oscillation region at expense of the high energy bias was considered to be an acceptable trade-off, therefore we decided to use training with the Flat Weight clipping.

7.2.7 LSTM EE Development: Sensitivity to the Major NOvA Systematics

After all performance optimizations of the LSTM EE, we got an energy estimator that achieves much better energy reconstruction at both NOvA detectors compared to the standard NOvA spline-based energy estimator. However, the LSTM energy estimator was trained and tested on the sample of simulated events.

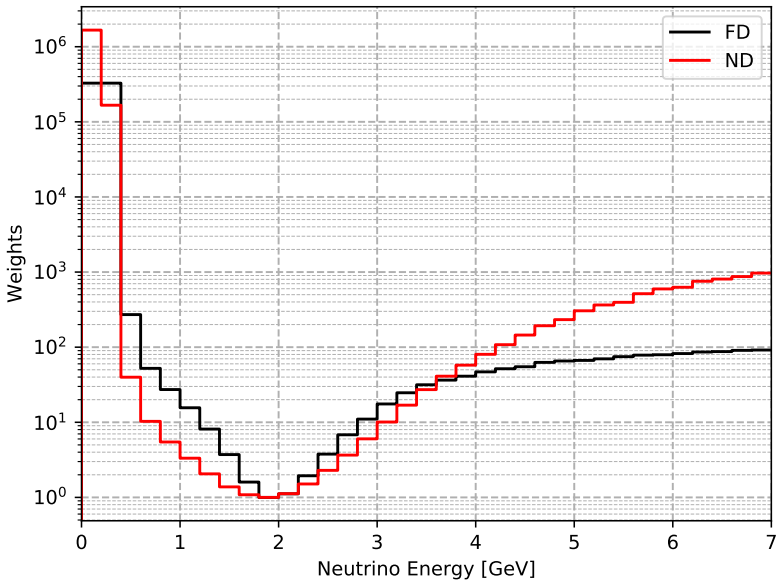


Figure 7.23: Comparison of the inverse of the true neutrino energy distributions between the NOvA Far Detector (black) and the Near Detector (red).

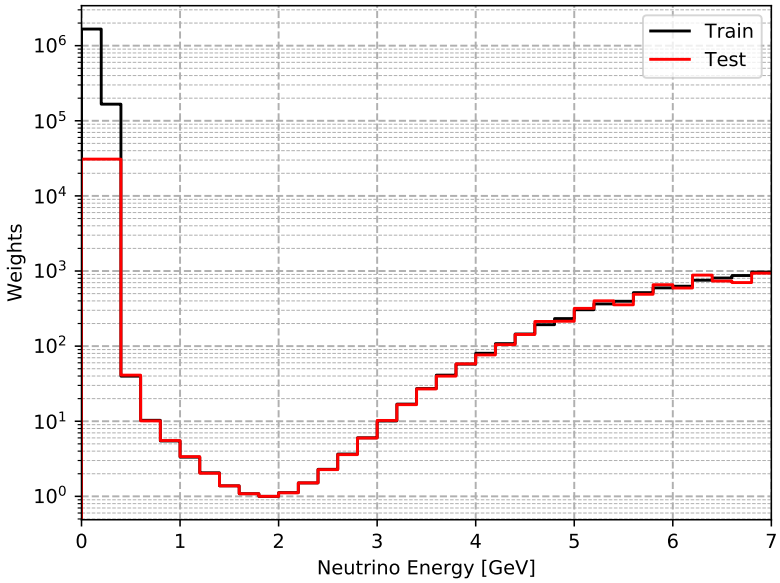


Figure 7.24: Comparison of the inverse of the true neutrino energy distributions between the training and test samples for the Near Detector.

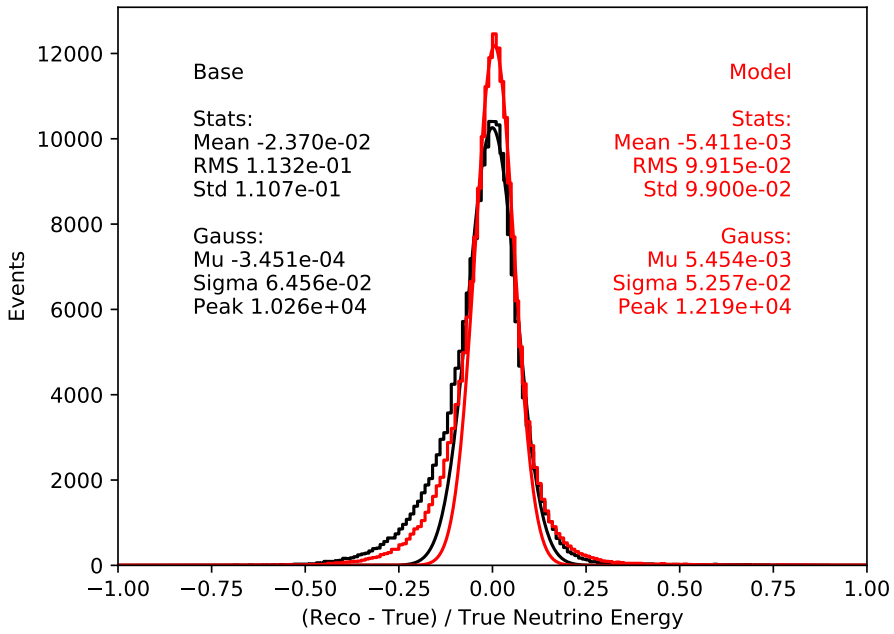
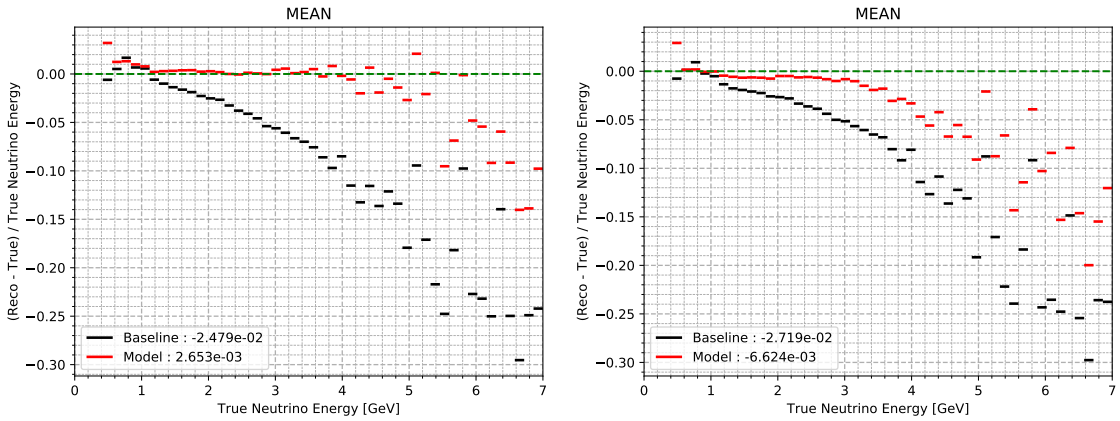


Figure 7.25: Near Detector energy resolution histograms for the spline-based energy estimator (black) and the LSTM energy estimator trained on the Loose Sample with the clipped Flat Weights (red). The energy estimators are evaluated on the Standard Sample.

If the NOvA MC model was exact, then we would be able to get better estimates of the oscillation parameters by using the LSTM energy estimator. And indeed, when we look at the sensitivity contours shown in Figure 7.27 for the neutrino oscillation parameters θ_{23} and Δm^2_{32}, we see that the contour obtained using the LSTM EE is better than the contour formed using the spline-based energy estimator.

Unfortunately, there are multiple uncertainties in the physics model that the NOvA experiment is using. If the LSTM energy estimator happens to be more sensitive to these uncertainties, then the gains brought by better energy reconstruction might be erased by larger error bands from the systematic uncertainties.

To understand the sensitivity of the LSTM energy estimator to various systematic uncertainties I have performed a number of studies [64], [65]. It turns out, that the LSTM EE has a sensitivity to the major NOvA systematics similar to the one of the standard NOvA spline-based energy estimator. Except for the Calibration



(a) Without Weight Clipping

(b) With Weight Clipping

Figure 7.26: Plots showing the means of the ν_μ energy resolution (Reco - True) / True plotted vs true neutrino energy bins. The red curves show the means of the LSTM energy estimator, while the black ones show the spline-based energy estimator. The LSTM EE was trained on the Loose Sample. The left plot shows the LSTM EE trained with the Flat Weights without clipping and the right plot shows the LSTM EE trained with the Flat Weights and with clipping. The energy estimators are evaluated on the Loose Sample.

systematics (that model calibration uncertainty of the calorimetric energies) – the LSTM EE is about twice as sensitive to the Calibration systematics compared to the spline-based energy estimator.

This can be illustrated by looking at how much the predicted energy distribution changes on a sample shifted by the Calibration systematics. The first plot of Figure 7.28 shows the ν_μ energy distributions predicted by the spline-based energy estimator. The black histogram is a histogram obtained on the nominal MC sample. The red histogram is obtained on the MC sample, shifted by the Calibration systematic by 5% up, and the blue one is shifted by 5% down. The second plot of Figure 7.28 shows the same distributions obtained with the LSTM energy estimator. Comparing the ratios of the ν_μ energy distributions between the Spline EE and the LSTM EE, we see that the LSTM energy distribution is more affected by the Calibration systematics.

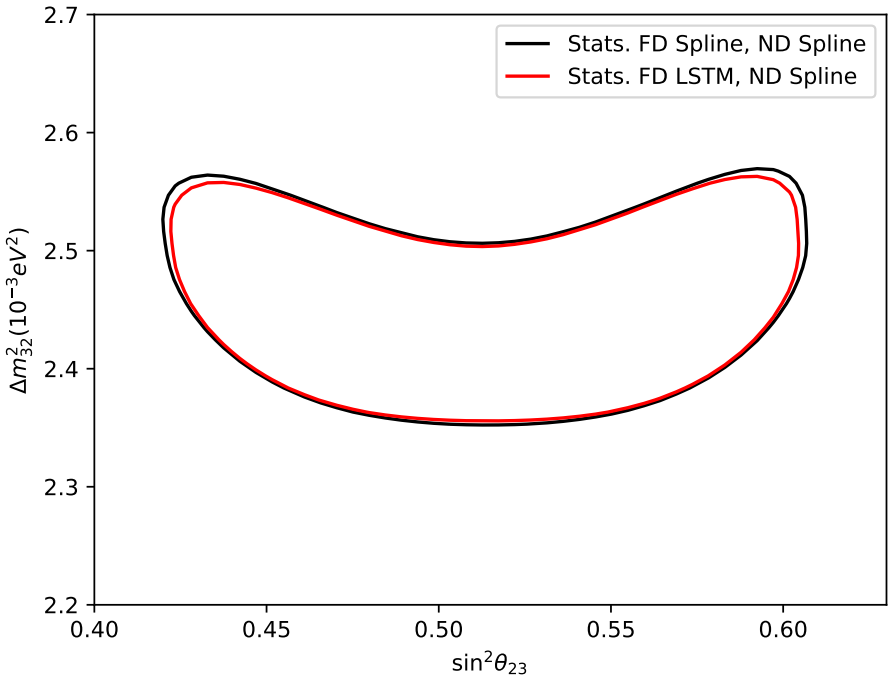
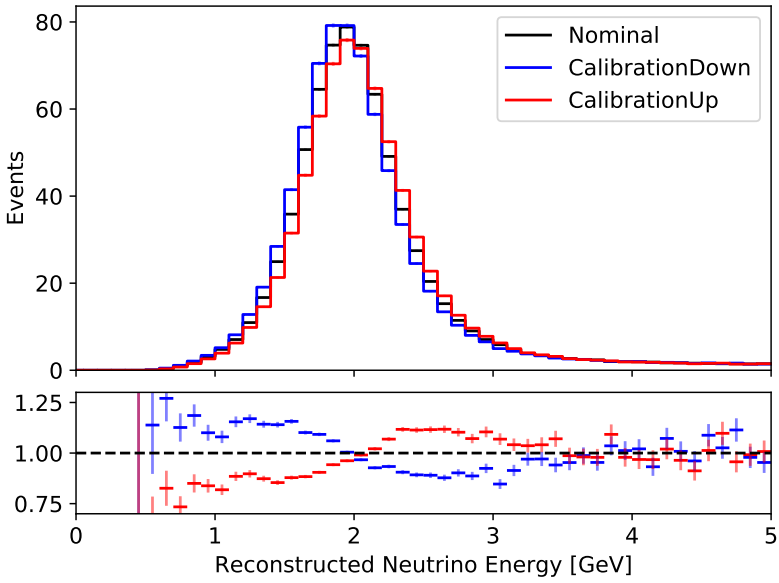
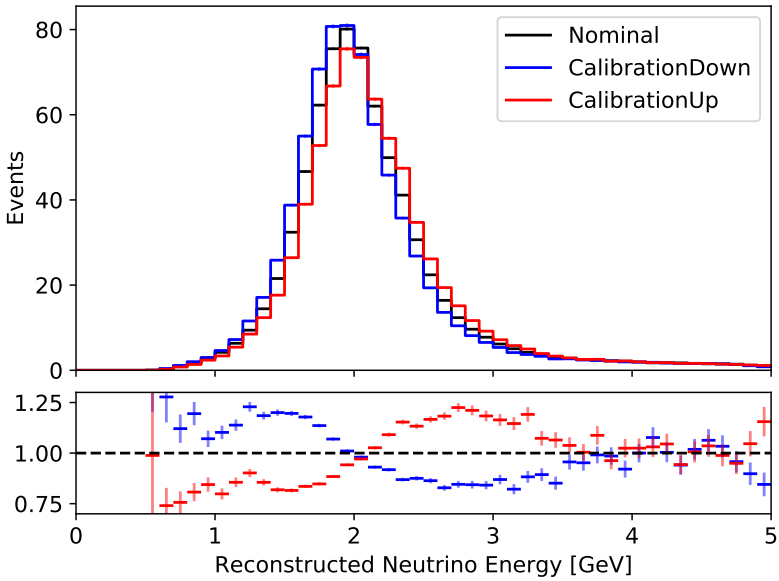


Figure 7.27: ν_μ only sensitivity contours in the Δm^2_{32}, $\sin^2\theta_{23}$ plane. The black contour is made with the standard NOvA spline-based energy estimator, while the red is the contour obtained with the LSTM energy estimator. Both contours are made under the assumption that there are no systematic uncertainties and they use only simulated Far Detector events from Period 3.



(a) Spline EE



(b) LSTM EE

Figure 7.28: Reconstructed neutrino energy distributions at the Far Detector. The black histogram is evaluated on the nominal FD Monte Carlo. The red histogram is evaluated on the FD MC shifted by the calibration systematic up, the blue one is shifted by the calibration systematic down. Below each histogram plot is a plot of ratios of the systematically shifted up/down histograms to the nominal one – it can be used to estimate the sensitivity of the energy estimator to the calibration shift.

The higher sensitivity of the LSTM energy estimator to the Calibration systematics does indeed erase all gains from the better energy reconstruction. Figure 7.29 shows ν_μ sensitivity contours made taking into account the calibration systematics. Comparing the LSTM EE contour to the Spline EE contour, we see that the LSTM EE gives worse estimates of the oscillation parameters due to higher sensitivity to the calibration systematics. Therefore, the next objective in the development of the LSTM energy estimator was to reduce its sensitivity to the Calibration systematics.

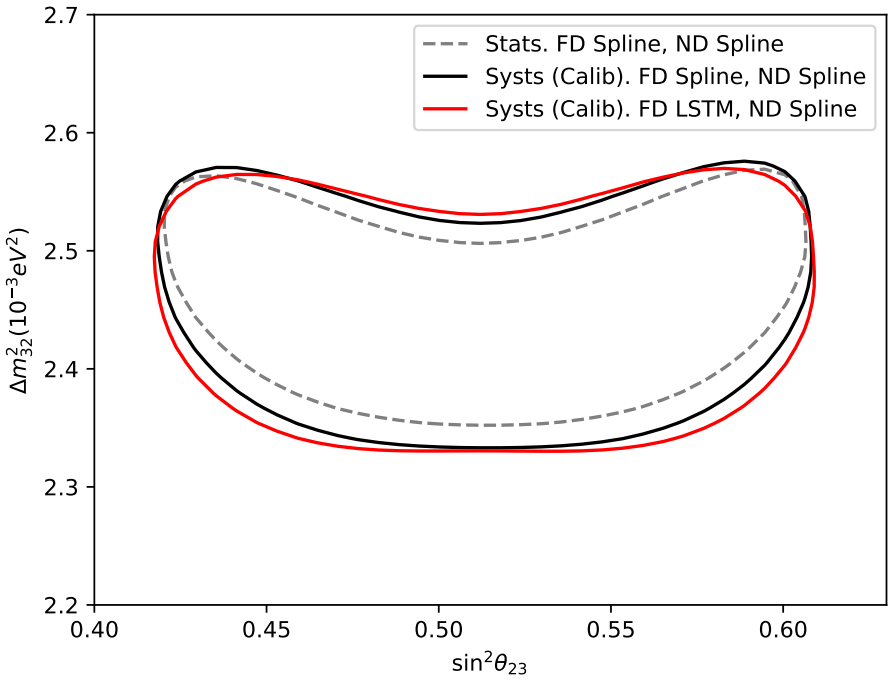


Figure 7.29: ν_μ only sensitivity contours in the Δm^2_{32}, $\sin^2\theta_{23}$ plane. The black contour is made with the standard NOvA spline-based energy estimator, while the red one is the contour obtained with the LSTM energy estimator. Both contours are made taking into account systematic uncertainties created by the calibration systematics only. The dashed contour is made with the spline-based energy estimator without any systematic uncertainties. These contours use the simulated Far Detector events from Period 3.

7.2.8 LSTM EE Development: Reduction of the Sensitivity to the Calibration Systematic

The calibration systematics are designed to model possible uncertainty of the calorimetric energies of the NOvA detectors. In order for the new energy estimator to be aware of this uncertainty, we need a way to transmit information that the calorimetric energies are not exact. This can be achieved, for example, by modifying the training procedure of the LSTM energy estimator and adding random shifts to the input variables, that are consistent with the action of the calibration systematic (i.e. uniform scaling of the calorimetric energy across the entire detector).

The list of all inputs that the LSTM energy estimator is using is presented in Table 7.2. Only a few of these inputs are directly affected by the calibration systematics. The affected inputs are

- `png.calE`, `png.weightedCalE` – calorimetric energies of prongs.
- `png2d.calE`, `png2d.weightedCalE` – calorimetric energies of 2D prongs.
- `png.bpf[i].overlapE` – overlap energies for BPF tracks.
- `calE`, `orphCalE`, `remPngCalE` – slice level information about calorimetric energies.

I have tried adding random shifts to these input variables during the training phase of the LSTM energy estimator. There are many ways such random shifts can be introduced and in particular, I have investigated three different random noise models that these shifts were sampled from:

- **Discrete Noise** – each time an event is seen during the training phase a random number ϵ is sampled from a set of $\{-a, 0, a\}$ with equal probability (33.(3)%). Then, all variables that are directly derived from the calorimetric energies are scaled by $1 + \epsilon$.
- **Uniform Noise** – is similar to the Discrete Noise, but the random number ϵ is drawn from the interval $[-a, a]$ with uniform probability.

- **Gaussian Noise** – same as before, but the random number ϵ is drawn from the normal distribution $\mathcal{N}(0, a)$.

The parameter a, introduced in the noise models described above, controls the magnitude of the noise and I have treated it as another hyperparameter of the LSTM energy estimator. Figure 7.30 shows how the sensitivity of the LSTM EE changes with the injection of the random noise to the calorimetric energies. According to Figure 7.30, the best performing noise model is Discrete and the minimal sensitivity to the Calibration systematic is achieved with $20 - 25\%$ noise. At the optimal noise value, the sensitivity of the LSTM EE to the Calibration systematic is more than 5 times smaller than the corresponding sensitivity of the standard spline-based EE.

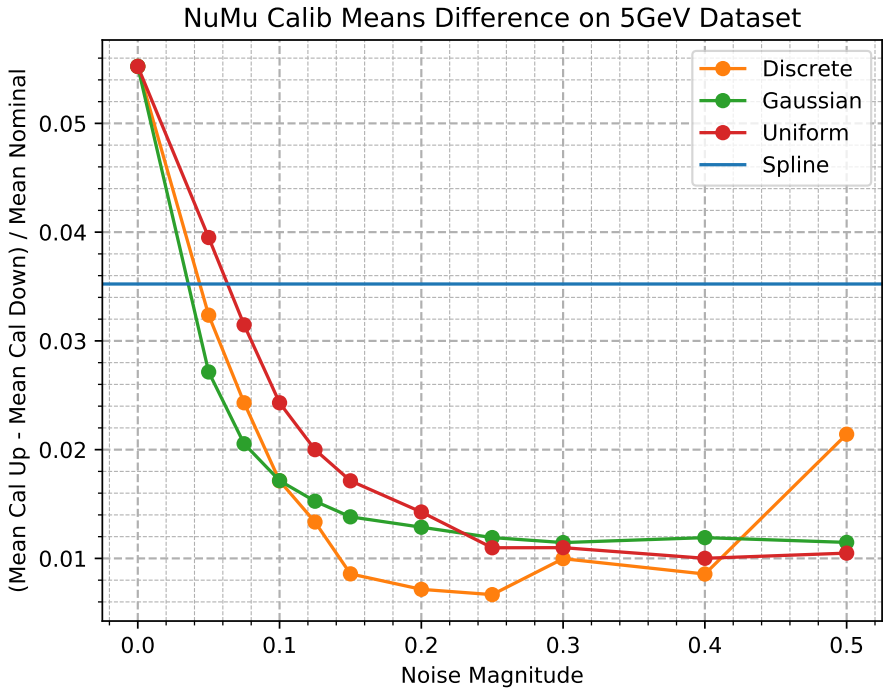


Figure 7.30: Sensitivity of the LSTM energy estimator to the Calibration systematic vs magnitude and shape of the noise that was added to the calorimetric energies during the LSTM EE training. The vertical axis shows the relative difference of the means of the NuMu energy predicted on an MC shifted up by the Calibration systematic minus the mean of the NuMu energy predicted on an MC shifted down by the Calibration systematic.

The addition of the noise to the calorimetric energies does decrease the energy resolution of the LSTM energy estimator, however, but not significantly (Figure 7.31). In other words, adding noise to the calorimetric energy inputs creates a trade-off between decreasing the sensitivity of the EE to the Calibration systematic on one hand and degrading the EE performance on the other hand. Ultimately, we are interested in the energy estimator that allows us to measure the neutrino oscillation parameters with the greatest precision. Therefore in order to find an optimal balance between EE sensitivity to the Calibration systematic and its performance I have made neutrino oscillation parameter contours, similar to Figure 7.29 with various noise configurations, and selected the one that gives the best precision. The results [66] show that the optimal performance is achieved with a 20% Discrete noise model.

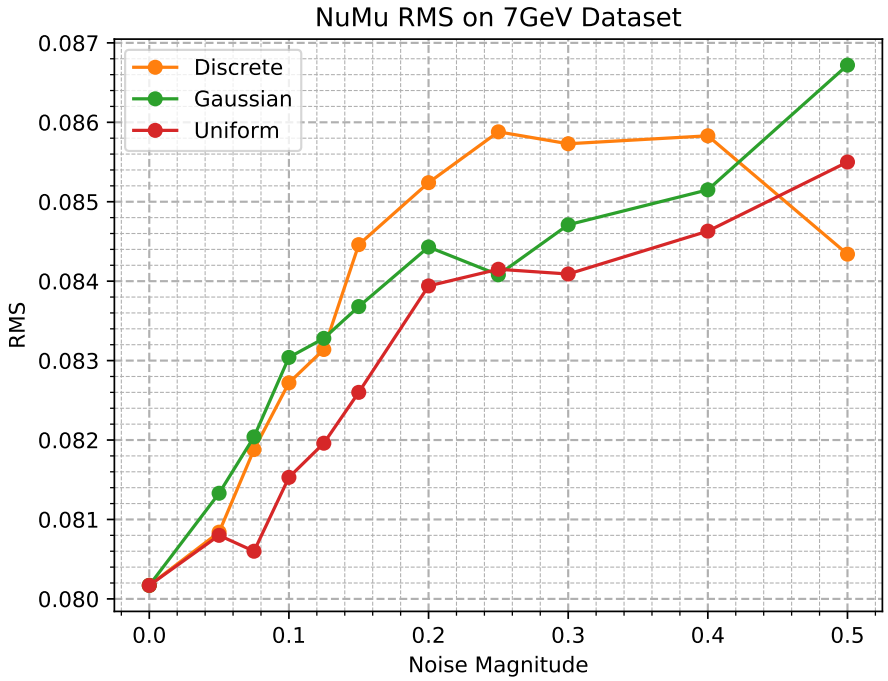


Figure 7.31: RMS of the neutrino energy resolution for the LSTM energy estimator vs magnitude and shape of the noise that was added to the calorimetric energies during the training phase.

Figure 7.32 shows the NuMu sensitivity contours obtained using the LSTM energy

estimator trained with 20% noise, overlaid with the contours made with the NOvA spline-based energy estimator. The LSTM EE exhibits much better performance than the old spline-based energy estimator.

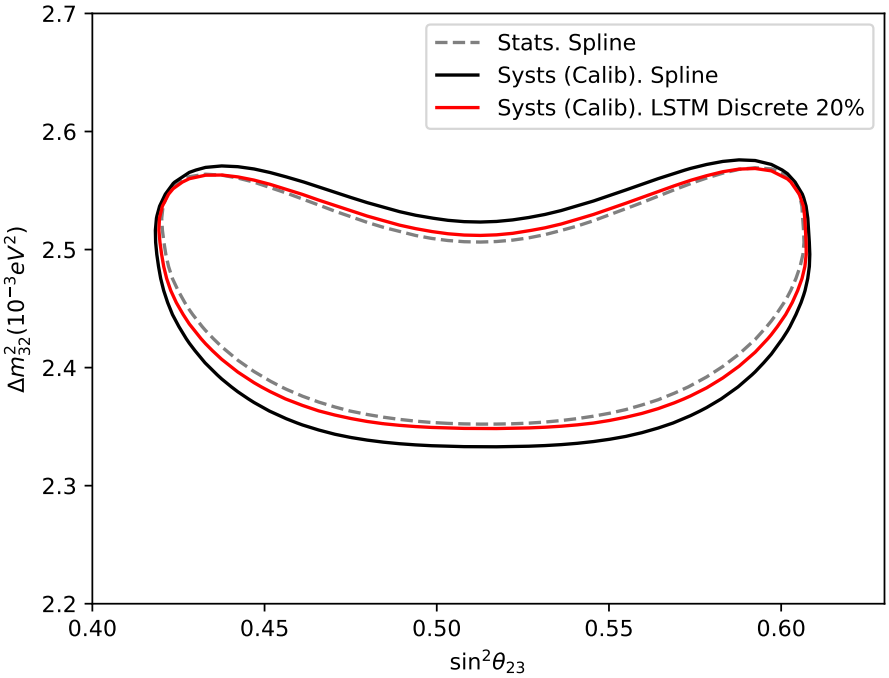


Figure 7.32: ν_μ only sensitivity contours in the Δm^2_{32}, $\sin^2\theta_{23}$ plane. The black contour is made with the standard NOvA spline-based energy estimator, while the red one is the contour obtained with the LSTM energy estimator with 20% Discrete noise. Both contours are made taking into account systematic uncertainties created by the calibration systematics only. The dashed contour is made with the spline-based energy estimator without any systematic uncertainties. These contours use the simulated Far Detector events from Period 3.

7.2.9 LSTM EE Development: Final Words

In this section, I have described the development of a new neutrino energy estimator that is based on a Recurrent Neural Network. This new energy estimator shows about 15% better energy reconstruction compared to the standard NOvA spline-based energy estimator (Figure 7.13), and it is more than 5 times less sensitive to

the major NOvA systematic uncertainty – the Calibration systematic (Figure 7.31). This energy estimator has been fully integrated into the NOvA ecosystem and the I have uploaded a python package to train this EE and study its performance, along with documentation to a github repository [1].

In its present state, the new energy estimator is ready to be used in the main NOvA analysis and I will expand on what performance the NOvA experiment can achieve using this new EE in chapter 8. There is one important issue currently remains unresolved, however. The issue is related to the number of input variables that the new energy estimator is using.

The LSTM EE is using about 50 different input variables that were summarized in Table 7.2, compared to just two input variables that the standard NOvA spline-based energy estimator is using. Since the LSTM EE uses a much larger number of input variables it is possible that NOvA currently lacks some systematics in order to model possible uncertainties of these inputs [67].

In particular, there is a growing concern that the NOvA experiment is relying on an incorrect model of secondary interactions (i.e. mismodeling by GEANT4 of interactions of hadrons with the detector in flight). Currently, there are undergoing efforts aimed at studying the effects of the mismodeling of secondary interactions. Unfortunately, until we get a clearer picture of these effects on the predicted value of the LSTM EE we cannot use it in the main analysis with confidence in results.

7.3 SliceLID Event Classifier

Another area where Machine Learning methods show a great deal of promise is event classification. The NOvA experiment already has one event interaction classifier (CVN) based on Convolutional Neural Networks [69]. This CVN event classifier uses raw event images in order to predict the type of the event, and it performs exceptionally well. However, since it is based on a convolutional neural network and operates on the raw data (hits in the detector) it is hard to interpret how this classifier operates [68]. Moreover, since various NOvA systematics operate on high-level physical quantities, it is even more difficult to study the behavior of the CVN

[1] `https://github.com/usert5432/lstm_ee`

classifier under such systematics, without completely resimulating events.

To address concerns about the black box-like behavior of the CVN classifier, it is useful to have a simpler event classifier, that operates directly on the physical quantities and that is easier to interpret.

Using the versatility of the Recurrent Neural Networks I have tried to build an event interaction type classifier (SliceLID) based on the LSTM neural cell. Similar to the LSTM Energy Estimator this event classifier uses prong and slice level input variables to make a prediction about the event interaction type for different event categories (ν_e-CC, ν_μ-CC, ν_τ-CC, NC, Cosmics).

Since this new event classifier (SliceLID) operates on the physical quantities as opposed to the raw detector images (like CVN), it will be much easier to understand how it operates and study its behavior under different systematics. On the other hand, transitioning from the event images to the reconstructed physical quantities we are going to lose some information. Therefore, on the theoretical grounds, the SliceLID classifier will never be able to surpass the performance of the CNN-based classifier. The purpose of SliceLID event classifier development is to use it as a cross-check to the event CVN classifier.

7.3.1 SliceLID Development: Initial Studies

The original SliceLID event classifier was developed by Andrew Vold[2] as a part of his Ph.D. thesis [70]. This version of SliceLID was capable only of predicting whether a given event is of ν_e-CC type or not. Its architecture was just a single LSTM cell that collected and processed prong level information to make an inference about the interaction type (Figure 7.33).

The original version of SliceLID relied on a number of input variables summarized in Table 7.3 in order to predict the event interaction type. As a slice level input, this classifier uses only the total calorimetric energy of the event. Therefore, the majority of the event type information is extracted from the prong level variables, which can be separated into several groups:

- `png.shwlid.lid.e{mu,g,pi,pi0,p,n}` – relative log-likelihoods indicating whether

[2]The code is published in the github repository: `https://github.com/andrew1102/SliceLID`

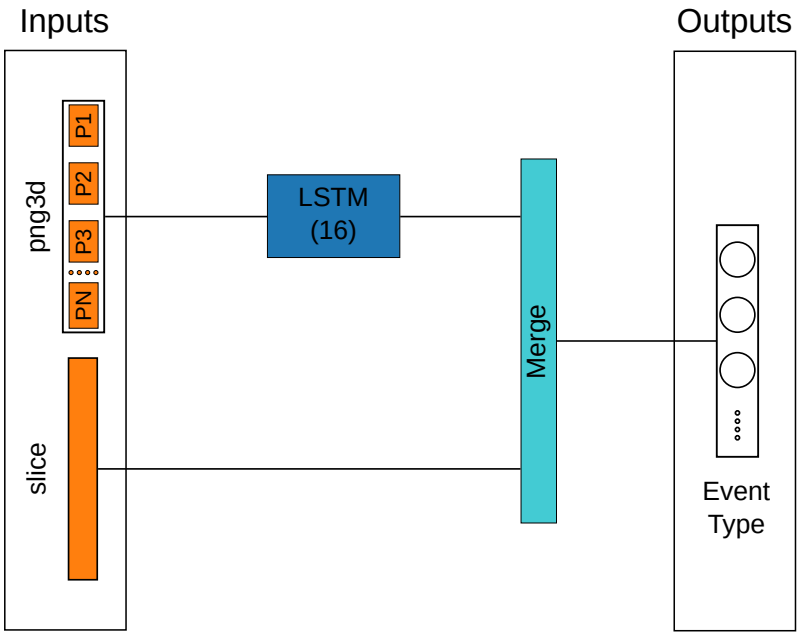


Figure 7.33: Original SliceLID classifier architecture.

the given prong was produced by a μ, γ, $\pi^{\pm}$, π^{0}, p, n or e. These log-likelihoods were computed by the ShowerLID module specially designed to analyze showers.

- `png.cvnpart.{muon,pion,proton,electron,photonid}` – scores indicating the likelihood of the given prong to be produced either by μ, $\pi^{\pm}$, p, e, or γ, predicted by Prong CVN.
- Prong length, calorimetric energy, and direction computed by the ShowerLID module.
- A gap between the interaction vertex and the first hit in the prong.
- Finally, the number of hits and the number of planes that the given prong spawns, also computed by the ShowerLID module.

I have used this model as a basis for the SliceLID development. First, I have trained it for distinguishing the ν_e-CC events from the other event types on a sample made of (ν_e-CC, ν_μ-CC, ν_τ-CC, NC) events. We can evaluate its baseline performance

3D prong vars:	**Slice vars:**

`vtx.elastic[0].fuzzyk.png:`
`png.cvnpart.{`
`  muon,pion,proton,`
`  electron,photonid`
`}`
`png.shwlid.{nhit,nhitx,hity,nplanex,nplaney}`
`png.shwlid.calE`
`png.shwlid.dir.{x,y,z}`
`png.shwlid.len`
`png.shwlid.gap`
`png.shwlid.lid.e{mu,g,pi,pi0,p,n}ll{l,t}`
	`calE`

Table 7.3: Summary of the inputs of the SliceLID classifier.

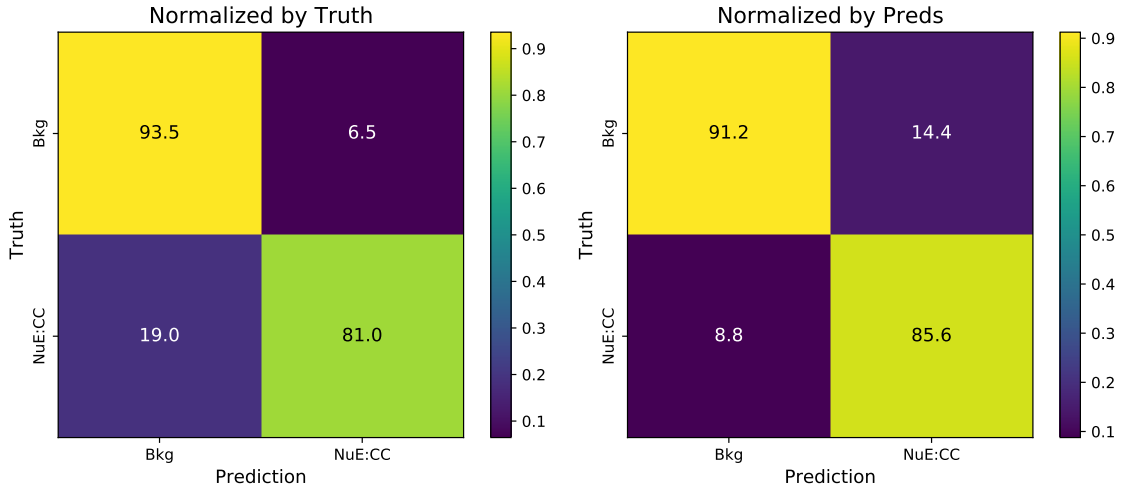
by looking at the error matrices shown in Figure 7.34 (also known as confusion matrices). The diagonal elements of these error matrices show the true positive/negative rates and the off-diagonal elements false positive/negative rates.

The error matrix on the left of Figure 7.34 shows rates normalized by truth (values in each row add up to 100%). The diagonal values of this error matrix show classification accuracy $= \frac{\{\text{True Component}\}\cap\{\text{Predicted Component}\}}{\{\text{True Component}\}}$ for a given component (also known as recall, or efficiency at NOvA). From the left plot of Figure 7.34, we can see that the original SliceLID classifier achieves 81% accuracy (also known as recall or efficiency) at classifying the ν_e-CC component. It also has 94% accuracy at classifying the background components (ν_μ-CC, ν_τ-CC, NC).

The right plot of Figure 7.34 shows the error matrices normalized by predictions (values add up to 100% in each column). Here, the diagonal elements indicate precision $= \frac{\{\text{True Component}\}\cap\{\text{Predicted Component}\}}{\{\text{Predicted Component}\}}$ (also known as purity at NOvA) of classification for each component. Reading these values we conclude that the original SliceLID classifier achieves 86% precision when classifying the ν_e-CC events and 91% precision when classifying the backgrounds.

7.3.2 SliceLID Development: Making Multitarget Classifier

The original SliceLID classifier is able to distinguish only the ν_e-CC component from the rest, which limits its application to the ν_e analysis only. However, we would like to have a multipurpose classifier that is also able to predict the likelihood that the



(a) Normalized by Truth

(b) Normalized by Prediction

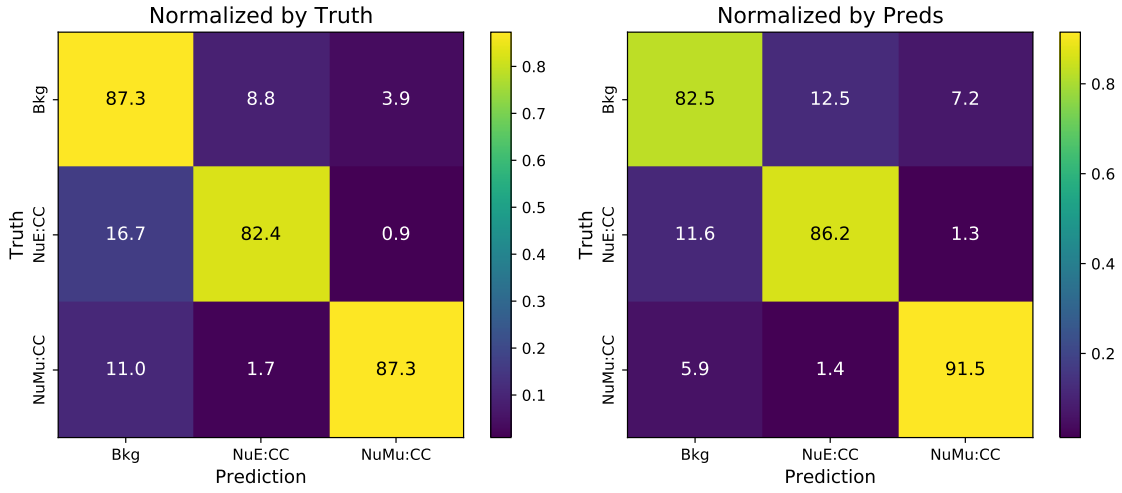
Figure 7.34: Error matrices for the original SliceLID classifier. The left plot shows the error matrix normalized by truth and the right one normalized by predictions.

given event is ν_μ-CC, or ν_τ-CC, or a cosmic ray.

To achieve this goal, I have tried to build a multitarget classifier, by iteratively adding additional target classes to the original architecture shown in Figure 7.33. I have started by separating the ν_μ-CC events into a separate class and training a 3-target classifier that is able to discriminate between ν_e-CC, ν_μ-CC, and background (ν_τ-CC and NC) events [71].

The performance of the resulting classifier is summarized in the error matrices shown in Figure 7.35. Judging by the error matrices we see that the ν_e component classification performance was largely unaffected (it became even slightly better due to a minor network optimization). The performance of the background rejection went down since the ν_μ-CC events were moved from the background component to their own class. The ν_μ-CC classification achieves a relatively high recall (87%) and precision (91%), which may hint to us that the task of classifying the ν_μ-CC events is relatively simple.

As a check for this performance, we can remember that a ν_μ-CC event has a distinctively long μ track and some hadronic activity, which is in stark contrast to ν_e-CC events that do not typically have any long tracks. Therefore it should be



(a) Normalized by Truth

(b) Normalized by Prediction

Figure 7.35: Error matrices for the 3-target SliceLID classifier. The left plot shows the error matrix normalized by truth and the right one normalized by predictions. The background component is made of ν_τ-CC and NC events.

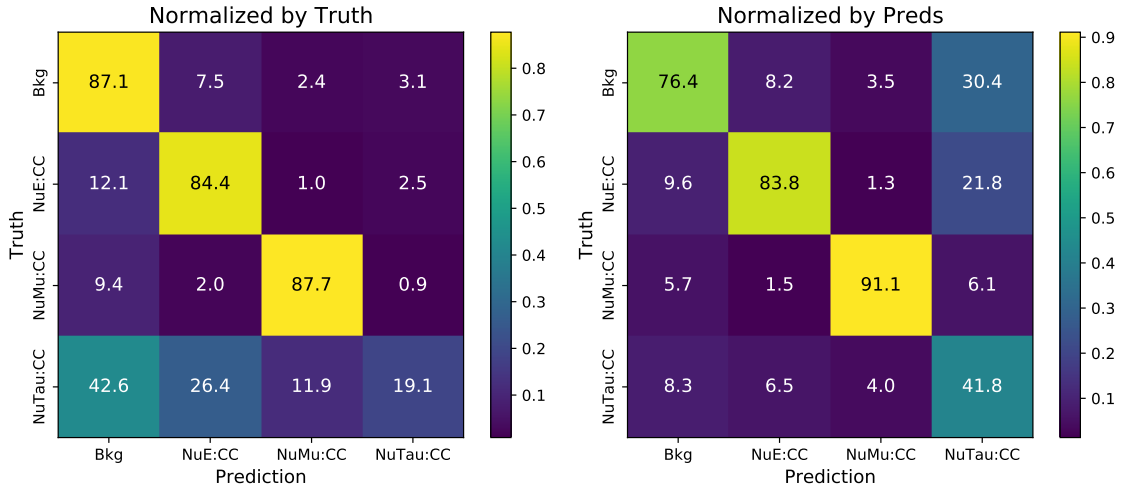
easy to distinguish them. And indeed, the off-diagonal components of Figure 7.35 between the ν_e-CC and ν_μ-CC events are virtually zero, i.e. the network makes little error at separating these types of events from each other.

7.3.3 SliceLID Development: Classifying ν_τ-CC events

After the ν_μ-CC events have been successfully incorporated into the SliceLID classifier, I have tried adding support for the classification of the ν_τ-CC events as well. Following the same approach as for the ν_μ-CC events, I have added a new ν_τ-CC class and retrained the classifier [72].

Figure 7.36 shows the error matrices for the SliceLID classifier trained with the addition of the ν_τ-CC component. The performance of SliceLID remained roughly unchanged for the ν_e-CC and ν_μ-CC events compared to the performance of the 3-class SliceLID classifier (Figure 7.35). However, the performance of the SliceLID for the ν_τ-CC events is rather poor, having a recall of about 19% and a precision of about 42%. Such a low performance is not acceptable for an event classifier.

I have tried to find the reason for this low performance. It turns out that the



(a) Normalized by Truth (b) Normalized by Prediction

Figure 7.36: Error matrices for the 4-target SliceLID classifier. The left plot shows the error matrix normalized by truth and the right one normalized by predictions. The background component is made of NC events.

NOvA MC has very a small proportion of the ν_τ-CC events compared to the other event classes (Figure 7.37). This is likely due to the fact that in the ν_τ-CC events a τ lepton should be produced, but it has a rather large mass of about 1.8 GeV. About half of the neutrinos in the training sample simply do not have enough energy to produce τ, so they cannot create a ν_τ-CC event.

To resolve this issue a couple of strategies can be employed. One way would be to generate a larger Monte Carlo sample of the ν_τ events. However, this was not possible due to the production constraints. Another way to work around this problem would be to upsample the ν_τ-CC events. This approach was deemed to be too complex to be undertaken given the time constraints for the SliceLID development. Correspondingly, I have opted for simply reweighting different event classes so that the resulting reweighted sample will have an equal contribution for each event type.

Figure 7.38 shows the result of retraining the SliceLID classifier on a reweighted sample. Comparing its performance to the SliceLID trained on the original sample (Figure 7.36), we can see that the recall for the ν_τ-CC events has improved signifi-

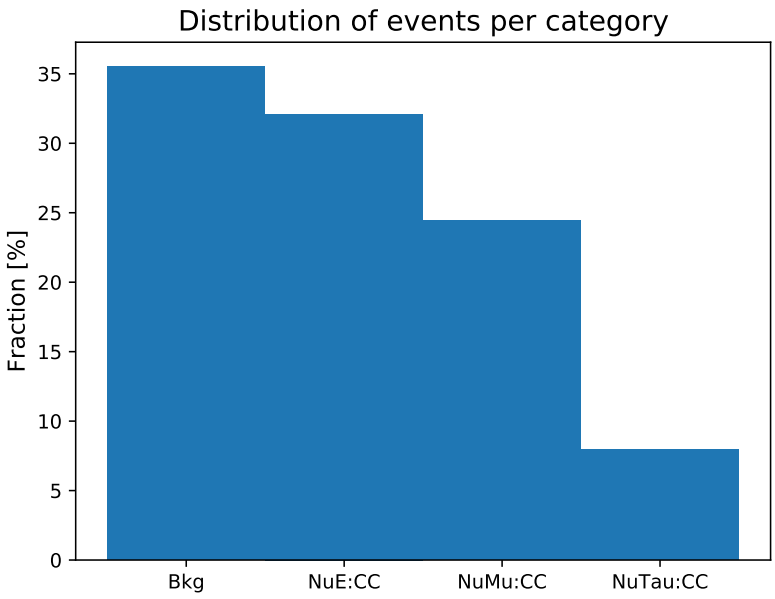
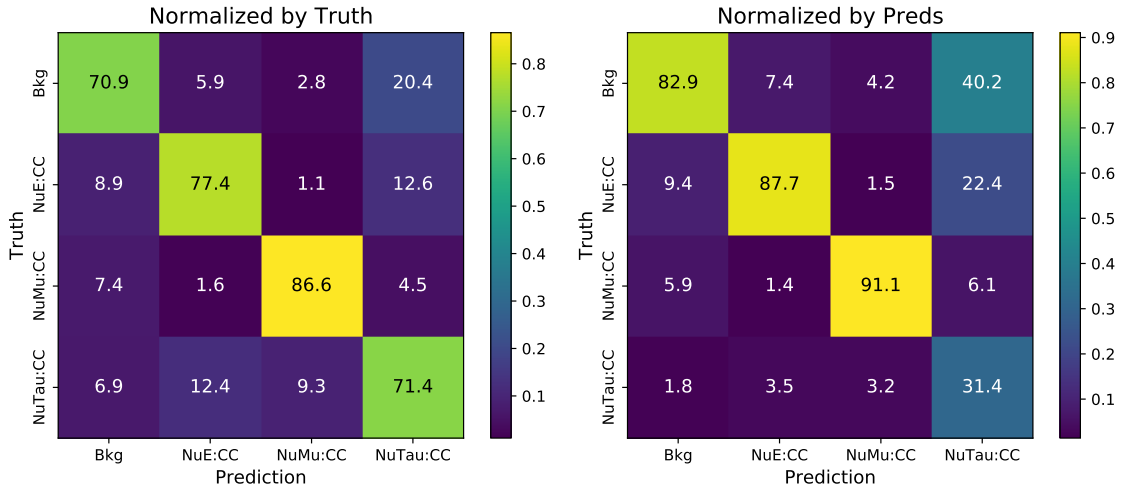


Figure 7.37: Distribution of events per target category in the 4-class SliceLID training sample.



(a) Normalized by Truth

(b) Normalized by Prediction

Figure 7.38: Error matrices for the 4-target SliceLID classifier trained with event class normalization. The left plot shows the error matrix normalized by truth and the right one normalized by predictions. The background component is made of NC events.

cantly from 19% to 71%. However, this improvement came at a cost of lowering the recall for the NC events (from 87% to 71%) and for the ν_e-CC events (from 84% to 77%). On the other hand, the classification precision decreased for the ν_τ-CC events and increased for the NC and ν_e-CC components.

The performance of the ν_μ-CC classification remained unaffected by the addition of the class weights. Looking at the off-diagonal components of Figure 7.38 we can see that there is a significant degree of misclassification between the ν_τ-CC, NC, and ν_e-CC events. This implies that the ν_τ-CC events have a high degree of resemblance to the NC and ν_e-CC events, but distinctively different from the ν_μ-CC events.

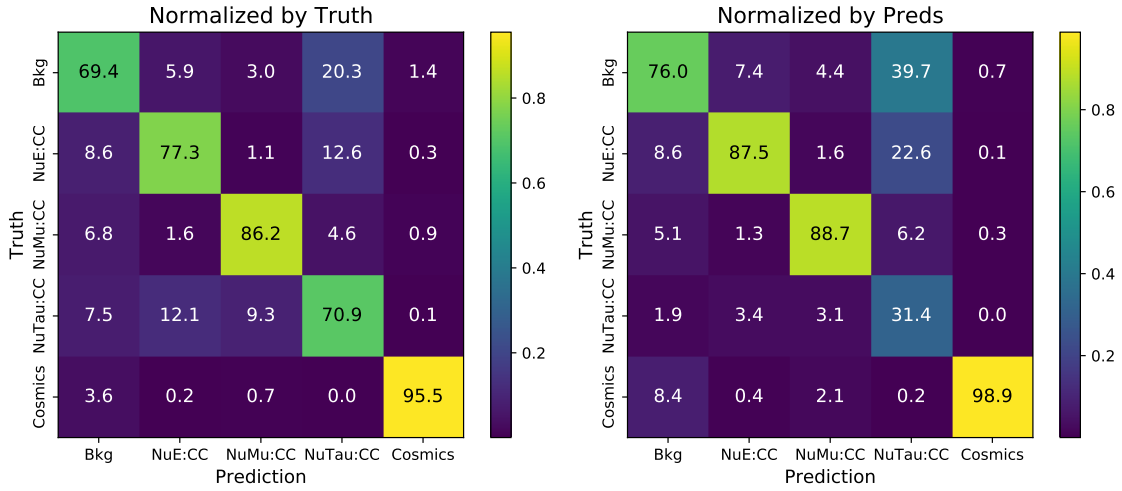
Judging by these observations it looks like the task of classification of the ν_τ-CC events is complex, since the network is incapable of separating them, despite the various approaches I have tried. It is possible that the addition of extra inputs and/or increasing the complexity of the model might improve the ν_τ-CC event classification. However, I was unable to identify extra input variables relevant to the ν_τ-CC classification task and the influence of model complexity on the SliceLID performance will be discussed in the following sections.

7.3.4 SliceLID Development: Classifying Cosmic Events

Finally, we would want the SliceLID classifier to be able to distinguish cosmic events. Therefore, I have tried adding cosmics using the same procedure as in the previous section and keeping the class normalization [73]. The error matrices for the resulting classifier are shown in Figure 7.39.

Similar to the addition of the ν_μ-CC events, the incorporation of the Cosmic events to the SliceLID classifier was smooth. Moreover, SliceLID achieved exceptional performance for the cosmic events, classifying them with 96% recall and 99% precision, while keeping the classification performance of the other components virtually unaffected.

At that moment the SliceLID classifier was capable of predicting all the event types required by its design. Next, I have tried to improve its performance by tuning the neural network architecture used for classification.



(a) Normalized by Truth (b) Normalized by Prediction

Figure 7.39: Error matrices for the SliceLID classifier trained with event class normalization. The left plot shows the error matrix normalized by truth and the right one normalized by predictions. The background is made of NC events.

7.3.5 SliceLID Development: Network Architecture Tuning

The original SliceLID network architecture (Figure 7.33) is very simplistic and there are a number of ways it can be improved. I have made a comprehensive study of network modification possibilities [72], [74]. During these investigations I have discovered that the network performance responds positively to increasing its complexity, however, the degree of improvement is rather small.

The studies of model complexity [74] show that the optimal performance of the SliceLID classifier is achieved with a network similar to the LSTM EE one (Figure 7.9). The final model that I have constructed for the SliceLID classifier is shown in Figure 7.40. Similar to the LSTM EE case it has large pre and post-processing branches around the LSTM cell. However, unlike the LSTM EE, it lacks the 2D prongs branch, since the 2D prongs do not have any useful PID variables that can help classification.

Using architecture from Figure 7.40 for the SliceLID classifier I have retrained it on the sample with equal class reweighting. The error matrices for the tuned SliceLID classifier are shown in Figure 7.41. Comparing these error matrices to the

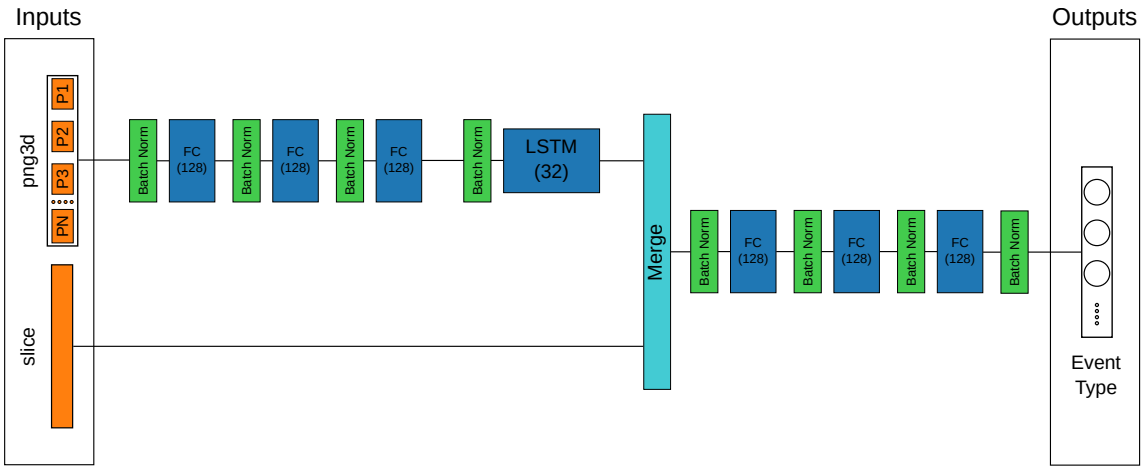
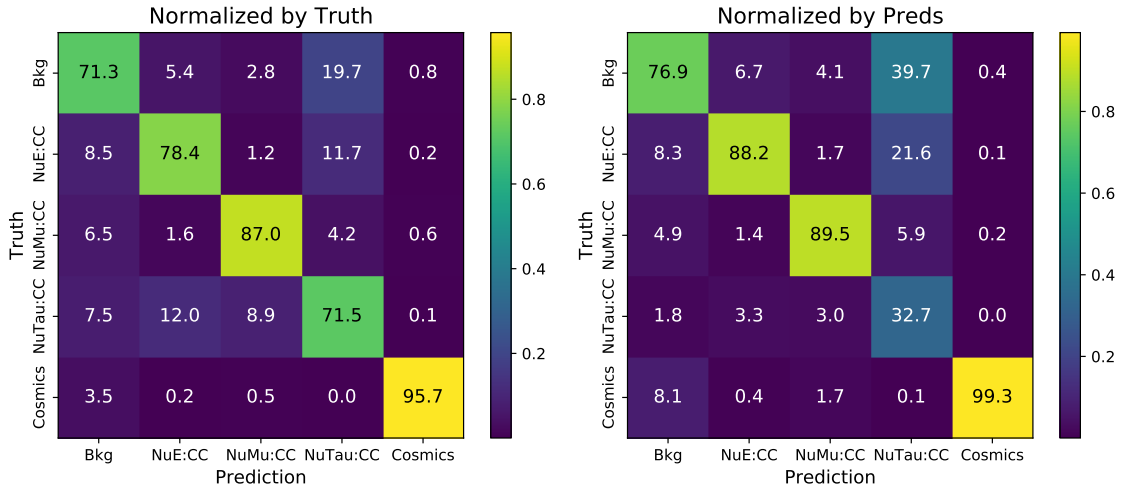


Figure 7.40: Tuned SliceLID classifier architecture.

error matrices of the original SliceLID classifier (Figure 7.39), we see an improvement in its performance across all event types. However, the improvement is of an order of about 1%.



(a) Normalized by Truth

(b) Normalized by Prediction

Figure 7.41: Error matrices for the tuned SliceLID classifier trained with event class normalization. The left plot shows the error matrix normalized by truth and the right one normalized by predictions. The background component is made of NC events.

The smallness of the improvement likely indicates that the input variables of SliceLID already contain useful information for the classification and further processing of this information produces little gain. Therefore, I believe that if significant improvement in the SliceLID performance is possible it will likely come from using additional relevant inputs for the event classification.

It is also worth noticing that making the SliceLID model more complex did not help significantly the ν_τ-CC events classification. Given that the NOvA analysis does not use the ν_τ-CC events, it may be possible to eliminate the ν_τ-CC classification from the goals of SliceLID and use the freed SliceLID capacity for better classifying the other event types.

7.3.6 SliceLID Performance

In the previous sections, I have described the development of the SliceLID event classification algorithm and studied its performance in terms of the error matrices (Figure 7.41). There are multiple other ways the performance of a classifier can be investigated. In this section, I will look at the t-SNE embedding plots of the SliceLID outputs [75]. The t-SNE embedding allows us to visualize clusters of events that the SliceLID classifier finds similar to each other. Observing such embeddings might give an intuition about what kinds of events SliceLID has difficulty separating from each other.

As we have seen in Figure 7.41, SliceLID experiences difficulty in classifying the ν_τ-CC events. Therefore, I will first look at the SliceLID performance on a sample without such events.

A two-dimensional t-SNE embedding of the SliceLID outputs evaluated on a sample without ν_τ-CC events is shown in Figure 7.42. As expected from Figure 7.41, SliceLID is quite good at separating cosmic events (violet) from the remaining event categories. For the ν_μ (green) and ν_e (orange) CC events, there are a number of clusters outside of the center of Figure 7.42 that are well separated from each other, indicating that there are certain ν_μ and ν_e event topologies that SliceLID can differentiate from each other. On the other hand, the NC events form one large continuous cluster in the middle of the picture. This implies that most NC events look very similar to each other from the point of view of the SliceLID classifier. Right at the

center of Figure 7.42, the NC event cluster mixes with the ν_μ and ν_e-CC events. This indicates that some fraction of the ν_μ and ν_e-CC events look very close to the NC events from the point of view of SliceLID. This similarity may be either due to SliceLID being not capable of distinguishing them (because of lack of relevant inputs or insufficient model expressiveness), or maybe due to the fact, that they are producing similar detector responses.

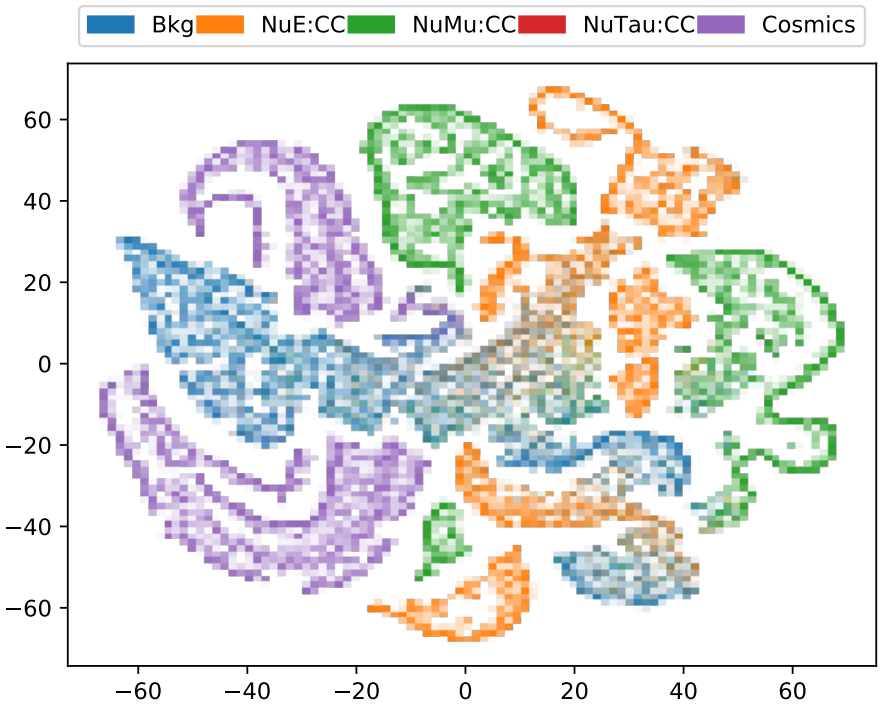


Figure 7.42: Two-dimensional t-SNE embedding of the SliceLID outputs obtained on a sample without ν_τ-CC events. The colors represent the true categories of the events. The brightness of each bin is proportional to the number of events in it.

Figure 7.43 shows a two-dimensional t-SNE embedding of the SliceLID outputs obtained on a sample with all event types. It differs from the previous plot by the presence of the ν_τ-CC events (red). The ν_τ-CC events have several topologies that SliceLID can differentiate between each other (represented by different red clusters in Figure 7.43). However, while SliceLID can differentiate these clusters from each other, it experiences a lot of difficulties differentiating them from clusters of events of other types. This is evident by the degree of mixing of the ν_τ-CC events with

the other categories. The amount of mixing between the ν_τ-CC events with the other ones may be another hint that probably a separate special purpose classifier is required for the ν_τ-CC events, and that SliceLID should be trained on a sample without them.

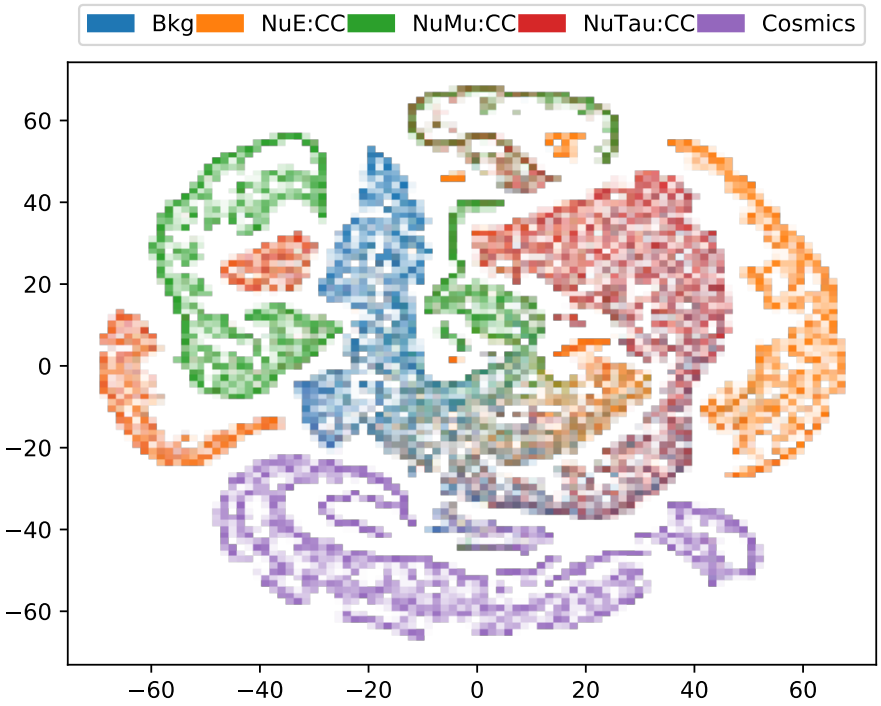


Figure 7.43: Two-dimensional t-SNE embedding of the SliceLID outputs. The colors represent the true categories of the events. The brightness of each bin is proportional to the number of events in it.

7.3.7 SliceLID Conclusion

In this section, I have described the development of the new event classifier SliceLID that is based on an architecture similar to the LSTM energy estimator. It achieves a reasonable classification accuracy (Figure 7.41) for the ν_μ-CC, ν_e-CC, NC, and cosmic events, but experiences large difficulties at recognizing the ν_τ-CC events.

In chapter 8, I will present further performance plots of the SliceLID classifier and discuss its possible applications for the main NOvA analysis.

Chapter 8

Results

I have described the development of the new neutrino energy estimator and the event classifier based on the LSTM neural network. In this section, I will show what improvements the NOvA experiment can get by using these algorithms.

8.1 LSTM Energy Estimator

I have tried to run the NOvA analysis chain similar to the official one that was used in the paper [76], but with the new LSTM energy estimator. However, as was mentioned in subsection 7.2.9 some of the NOvA systematic uncertainties will require significant modification in order to work with the LSTM EE, and multiple additional studies need to be done to find a proper way of making such modification. Until such studies are complete, I have decided to leave two systematics (out of about a hundred) out of consideration and redo the NOvA analysis without them.

The systematics that I will exclude from the analysis are the Muon Range uncertainty systematic and the Neutron systematic. The Muon Range uncertainty systematic tries to model various ionization effects in the detector's scintillator and corresponding changes in the length of the muon track. This systematic is required since the standard NOvA energy estimator uses the track length of a muon in order to predict its energy. The LSTM energy estimator, however, relies not only on the track length of the muon but also on the track lengths of all charged particles. Therefore, in order to extend the range systematic to the LSTM EE, we would need

to study the effects of multiple scintillator ionization properties on tracks of many charged particles. Before such studies are complete, I cannot reliably use the Range systematic with the LSTM EE and therefore I have excluded it from the analysis.

The Neutron systematic is a data-driven systematic that was designed to model calorimetric energy discrepancy between Data and MC for prongs of the particles that were created by neutrons [77]. It is required since the standard NOvA energy estimator relies on the total calorimetric energy of all hadronic activity in the detector. However, similar to the case of the muon range systematic, the LSTM EE relies not only on the calorimetric energy of hadrons but also on their track lengths, numbers of hits in each cluster, CVN scores, etc. In order to make the Neutron systematic work with the LSTM EE, we would need to study the Data/MC discrepancy for all input variables that the LSTM EE is using. Until such a study is complete, I am going to exclude the Neutron systematic from the analysis as well.

I have repeated the NOvA NuMu disappearance analysis using the standard NOvA energy estimator and the new LSTM energy estimator on the FHC dataset from period 3 without using cosmic files. The resulting 1σ neutrino oscillation parameter contours are shown in Figure 8.1 [78]. The contour obtained with the LSTM EE is narrower compared to the contour obtained with the old energy estimator, which means that the LSTM EE gives more precise results.

We can take a detailed look at various systematics and see how much uncertainty they contribute to the estimated oscillation parameters. Figure 8.2 shows magnitudes of uncertainties produced by various systematics for the Spline EE vs LSTM EE. The LSTM EE related systematic uncertainties are either smaller or equal to the systematic uncertainties produces by the Spline EE. The largest improvement is coming from the Hadronic Energy scales, which are modeled by the Calibration systematic. This is hardly surprising since I have taken special steps in order to reduce the influence of this systematic.

Finally, it is instructive to look at how much additional data would be needed to produce improvement in the neutrino oscillation parameter contours (Figure 8.1) with the old energy estimator in order to get performance comparable to simply using the new energy estimator (without extra data).

Figure 8.3 shows the NuMu sensitivity contours for the Spline and LSTM EEs,

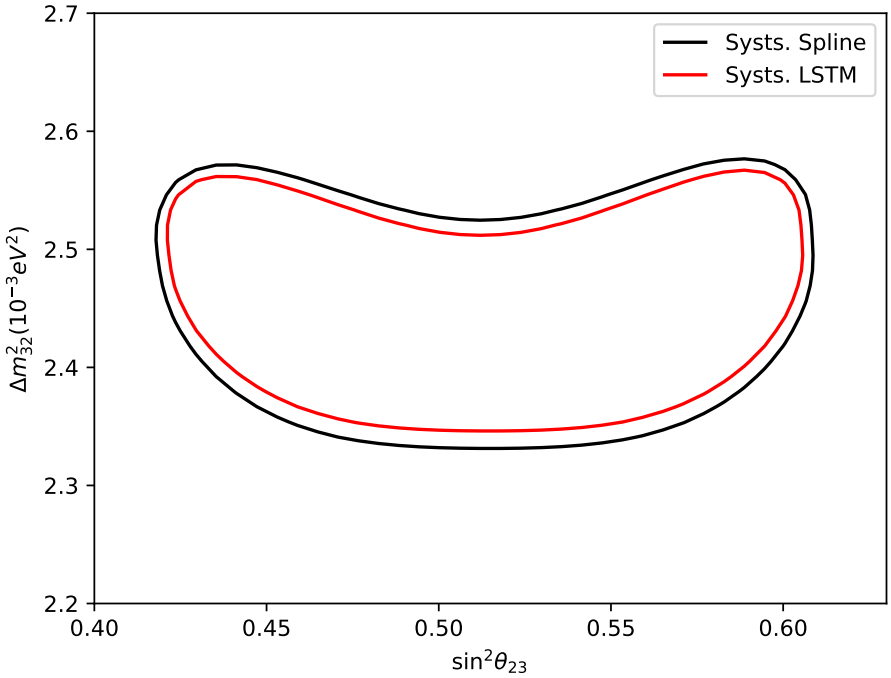


Figure 8.1: ν_μ only sensitivity contours in the Δm^2_{32}, $\sin^2\theta_{23}$ plane. The black contour is made with the standard NOvA spline-based energy estimator, while the red one is the contour obtained with the LSTM energy estimator.

with the Spline EE evaluated on datasets with a progressively increasing number of events. Focusing on the vertical parts of the contours we see, that the LSTM EE contour lies in between Spline EE contours with $10-20\%$ extra statistics. This would imply that usage of the new energy estimator is equivalent to running NOvA for $10-20\%$ more years with the old energy estimator for the $\sin^2\theta_{23}$ neutrino oscillation parameter.

On the other hand, if we focus on the horizontal parts of the contour on the Figure 8.3 then we see that when we increase the amount of data available to the Spline energy estimator its contours shrink, however even with 30% more data, the Spline EE contour still does not even touch the LSTM EE contour. This means that the usage of the LSTM energy estimator is equivalent to running NOvA for $30-50\%$ more years with the old energy estimator for the Δm^2_{32} neutrino oscillation parameter.

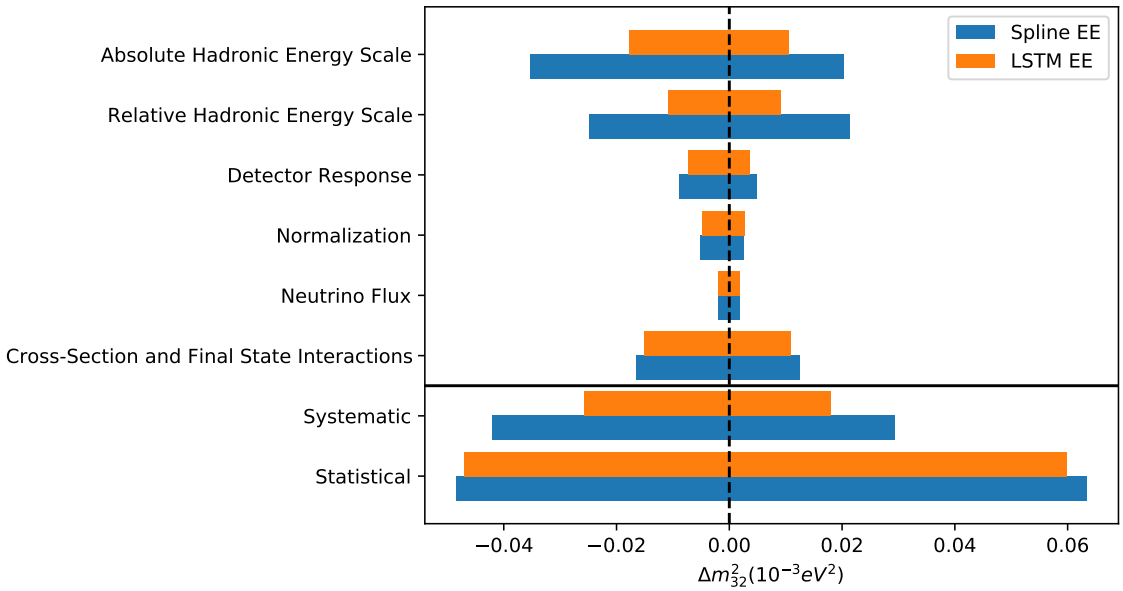


Figure 8.2: Summary of the two-sided uncertainties of the estimated Δm^2_{32} neutrino oscillation parameter. The blue bars represent uncertainties of the oscillation parameter obtained with the standard NOvA energy estimator and the orange bars represent uncertainties obtained with the LSTM EE. The horizontal labels in the top part show names of the systematics. The two bottom labels indicate the total systematic and total statistical uncertainties.

Judging by Figure 8.3 and Figure 8.2 the new LSTM energy estimator that I have developed significantly outperforms the standard NOvA energy estimator and allows us to achieve $10-50\%$ larger sensitivity to the neutrino oscillation parameters. However, as I have mentioned above there are two systematics that were omitted from the analysis – the Muon Range systematic and the Neutron systematic.

Preliminary studies [79] [80] of the Muon Range systematic indicate that the LSTM energy estimator is also less sensitive to this systematic than the standard spline-based EE. However, no studies have been performed for the Neutron systematic. We also do not know at this moment the uncertainty related to the mismodelling of hadronic interactions in flight. But, even if it is determined that the LSTM EE is more affected by these uncertainties compared to the Spline EE, we can reduce their effect by repeating the procedure described in subsection 7.2.8.

Apart from studying the effects of the remaining systematic uncertainties the

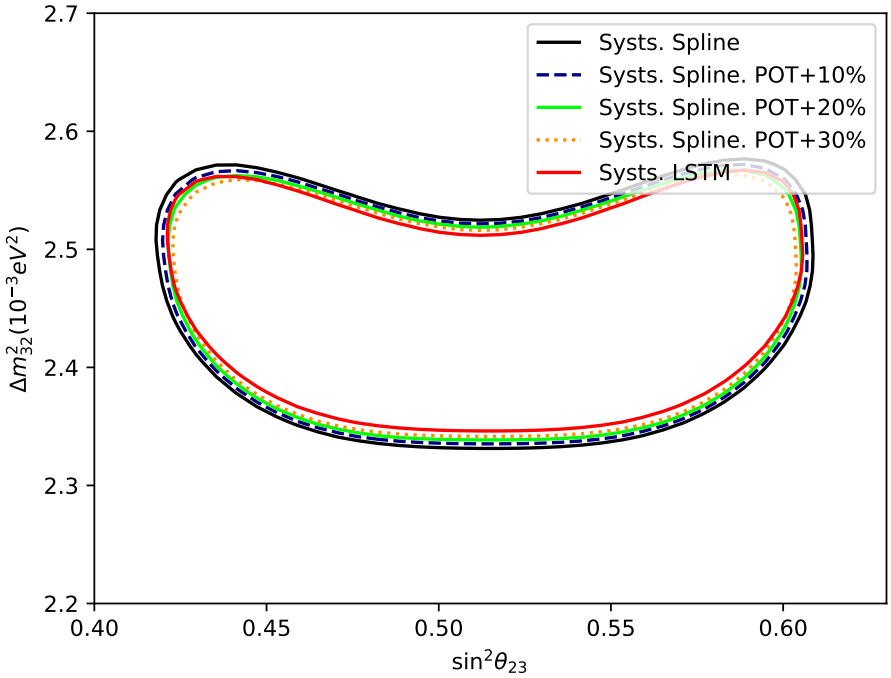


Figure 8.3: ν_μ only sensitivity contours in the Δm^2_{32}, $\sin^2\theta_{23}$ plane. The black contour is made with the standard NOvA spline-based energy estimator, while the red one is the contour obtained with the LSTM EE. The other contours are made with the Spline EE, but use 10%, 20%, and 30% more data.

development and deployment of the LSTM energy estimator have been finalized. The code required to evaluate the LSTM EE within the NOvA analysis framework has been committed and validated. The package required to train and evaluate the LSTM EE has also been committed to the github repository [1]. The documentation and the tutorial for the LSTM EE can be found on the NOvA docdb archive [81].

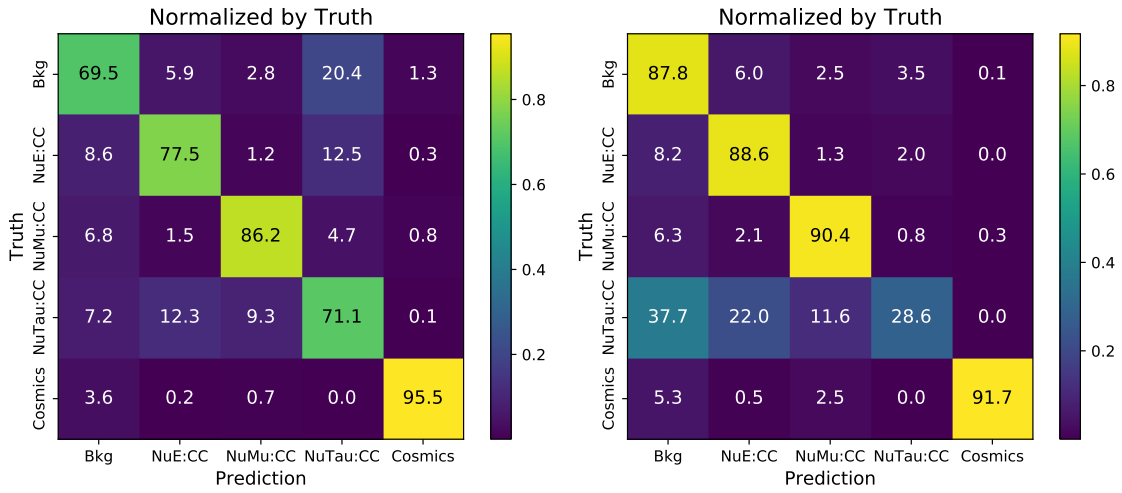
8.2 SliceLID

The SliceLID event classifier was developed with the primary purpose of being an interpretable alternative to the standard NOvA event classification method (CVN) that relies on a Convolutional Neural Network. As was mentioned previously, SliceLID

[1] `https://github.com/usert5432/lstm_ee`

receives a much smaller amount of information compared to the CVN and therefore theoretically its performance cannot exceed the performance of the CVN event classifier.

Nevertheless, it is instructive to compare the performance of SliceLID to the CVN directly in order to get an idea of how much information is lost when going from the complete image of neutrino event to only the high-level information about prongs.



(a) SliceLID (b) CVN

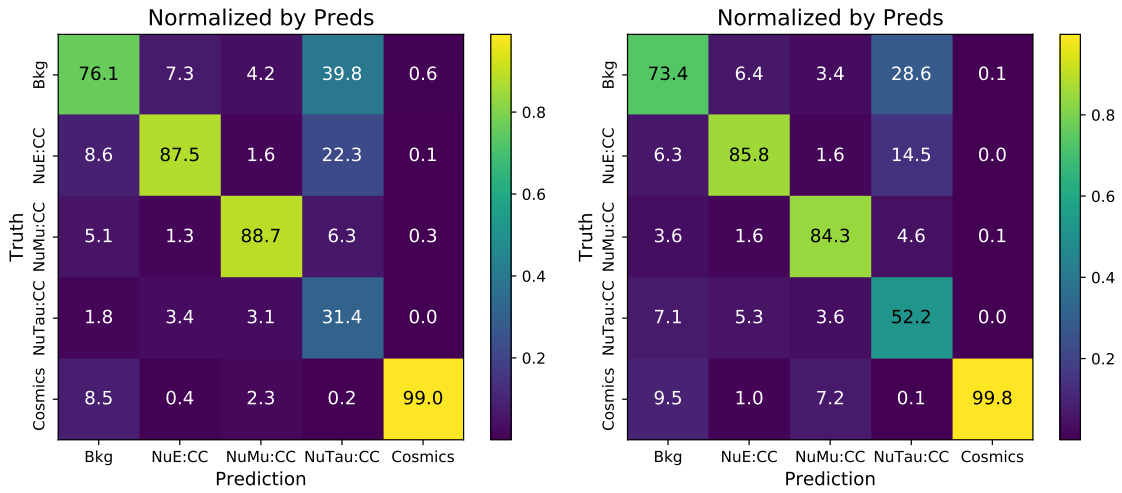
Figure 8.4: Classification error matrices normalized by truth (Efficiency). The error matrix on the left is obtained using the SliceLID classifier, the error matrix on the right is obtained using the CVN classifier.

Figure 8.4 [82] shows classification error matrices normalized by truth for SliceLID and CVN evaluated on the same Far Detector sample. Comparing these error matrices we see that CVN has a higher recall than SliceLID for NC, ν_μ-CC and ν_e-CC events, but a lower recall for the ν_τ-CC and Cosmic events.

I have been able to trace this discrepancy in classification results to the fact, that the CVN classifier was trained on a sample with imbalanced classes which led to the underrepresentation of the ν_τ-CC and cosmic events in the training sample. Unfortunately, the imbalance of classes in the CVN training sample prevents me from making a direct comparison between SliceLID and CVN.

Figure 8.5 shows classification error matrices normalized by predictions for SliceLID

and CVN. These plots show, that the SliceLID has higher classification precision for all event categories, except ν_τ-CC and Cosmic events. This picture is the opposite of Figure 8.4 and simply indicates the trade-off between precision and recall.



(a) SliceLID (b) CVN

Figure 8.5: Classification error matrices normalized by predictions (Purity). The error matrix on the left is obtained using the SliceLID classifier, the error matrix on the right is obtained using the CVN classifier.

While SliceLID was developed as just a more simple and interpretable version of CVN, the classification matrices Figure 8.4 and Figure 8.5 indicate that SliceLID may be also used as an auxiliary classifier in order to improve the classification of the ν_τ-CC and cosmic events.

Similar to the LSTM energy estimator, the code required to use the SliceLID classifier has been fully integrated into the NOvA software framework. The package to train and evaluate SliceLID has also been committed to the github repository [2]. I have also written documentation about how to train and use the SliceLID classifier which can be found on the nova docdb archive [83].

[2] `https://github.com/usert5432/slice_lid`

Chapter 9

Conclusions

In this thesis, I have discussed my work on the NOvA experiment related to the design and development of the novel neutrino energy estimator and event classifier based on Recurrent Neural Networks.

The new neutrino energy estimator LSTM EE that I have developed has $10-15\%$ better energy reconstruction performance compared to the standard NOvA energy estimator. It is also 5 times less sensitive to the major systematic uncertainty at NOvA. The usage of the new energy estimator with the standard analysis chain could allow NOvA to estimate neutrino oscillation parameters significantly better than with using the standard energy estimator. The magnitude of improvement with new energy estimator corresponds to having $10-50\%$ additional data with the old energy estimator (depending on the oscillation parameter in question). In other words, the adoption of the new energy estimator is equivalent to running the NOvA experiment for $10-50\%$ more time with the old one. However, in order to get a full confidence in the results obtained with the new energy estimator, additional studies of the systematic uncertainties need to be performed.

The new event classifier SliceLID was intended to be used only as an interpretable version of the standard NOvA event classification method with Convolutional Neural Networks. SliceLID has performance comparable to the standard NOvA event classifier, but it focuses on different event types. Therefore, in addition to being used simply as an interpretable version of the CNN classifier, it can be used as an auxiliary event classifier to better classify the event types that the CNN classifier

ignores.

Both, LSTM EE and SliceLID are the examples of a successful application of the Deep Learning techniques in the high energy physics, and I believe the NOvA experiment will benefit from usage of a larger number of the Deep Learning algorithms in its analysis.

Acknowledgment: This material is based upon work supported by the U.S. Department of Energy, Office of Science, Experimental Research at the Intensity Frontier in High Energy Physics program under Award Number DE-SC0012069.

Glossary

ADC Analog Digital Convertor that digitizes analog data from the APDs. This abbreviation is also used interchangably to denote APD values, digitized by ADC . 41, 47

APD Avalanche Photodiode. 35, 40, 41, 46, 51, 155

cut A filter that is used during the analysis phase in order to select signal events and reject background events . 79

CVN Convolutional Visual Network. An umbrella term used on NOvA to describe a class of Convolutional Neural Networks used for event and particle identification . viii, 70, 81, 82, 85, 86, 103, 132, 134, 147, 150

Data Used to denote the real detector data, as opposed to a sample of simulated detector events . 50, 51, 88, 147, 164

EE Shorthand for Energy Estimator. 146

FD Far Detector. 40, 86, 106, 117

FEB Front-End Board. An electronic board collecting data from 32 APDs of a cell extrusion module . 41, 46

FHC Forward Horn Current mode of the NuMI beam. In this mode the NuMI beam is primarily made of muon neutrions . xvi, 27, 37, 147, 166, 167

MC Monte Carlo. Used to denote simulated sample of detector events . 50, 51, 86, 88, 95, 123, 138, 147, 164

ND Near Detector. 40, 87, 106, 117

PID Particle Identification or Particle Identifier. Used to denote either algorithm responsible for prediction of the particle type or the result of such prediction . 78, 79, 109, 141

prong A cluster of hits that were grouped together by an algorithm, aimed to identify individual particles in each event (c.f. subsection 5.1.4) . 60, 63, 65, 79, 80, 102, 133, 147, 151

RHC Reverse Horn Current mode of the NuMI beam. In this mode the NuMI beam is primarily made of muon antineutrions . 27, 37

slice A cluster of detector hits that is intended to contain all hits produced in a single detector interaction. Such cluster is found by the slicing algorithm described in subsection 5.1.1 . 60, 69, 103, 133

spill A short period of time ($\approx 8-10\,\mu$s) corresponding to protons bombarding the NuMI target. The NuMI (anti-) neutrinos are produced and detected during these windows. subsection 3.2.1 . 27, 36

Spline EE Spline Energy Estimator. The standard spline-based ν_μ-CC energy estimator used in the NOvA analysis (c.f. subsection 5.5.1) . 147, 166

Bibliography

[1] Pierre Ramond, Introduction to neutrinos arXiv:hep-ph/0001007

[2] Gelmini, G. B. et al, Through Neutrino Eyes: Ghostly Particles Become Astronomical Tools, DOI:10.1038/scientificamerican0510-38

[3] Sacha Davidson, Enrico Nardi, Yosef Nir, Leptogenesis, 10.1016/j.physrep.2008.06.002

[4] Michael E. Peskin, Daniel V. Schroeder, An Introduction to Quantum Field Theory, ISBN-13: 978-0201503975

[5] Jean-Michel Levy, On the derivation of the neutrino oscillation length formula, arXiv:hep-ph/0004221

[6] Giganti, Claudio, Stéphane Lavignac, and Marco Zito, Neutrino oscillations: the rise of the PMNS paradigm, Progress in Particle and Nuclear Physics 98 (2018): 1-54. DOI: 10.1016/j.ppnp.2017.10.001

[7] M. Maltoni, T. Schwetz, M. A. Tórtola, and J. W. F. Valle, Constraining neutrino oscillation parameters with current solar and atmospheric data, Phys. Rev. D 67, 013011, DOI: 10.1103/PhysRevD.67.013011

[8] The Super-Kamiokande Collaboration, Evidence for oscillation of atmospheric neutrinos, Phys. Rev. Lett. 81, 1562, DOI: 10.1103/PhysRevLett.81.1562

[9] D. V. Forero, M. Tórtola, and J. W. F. Valle, Global status of neutrino oscillation parameters after Neutrino-2012, Phys. Rev. D 86, 073012, DOI: 10.1103/PhysRevD.86.073012

[10] The KamLAND Collaboration, Reactor On-Off Antineutrino Measurement with KamLAND, arXiv:1303.4667

[11] Daya Bay Collaboration, New Measurement of Antineutrino Oscillation with the Full Detector Configuration at Daya Bay, Phys. Rev. Lett. 115, 111802, DOI: 10.1103/PhysRevLett.115.111802

[12] NOvA Collaboration, The NOvA Technical Design Report, 2007, DOI: 10.2172/935497

[13] G. J. Feldman, J. Hartnell, T. Kobayashi, A Review of Long-baseline Neutrino Oscillation Experiments, 2013, DOI: 10.1155/2013/475749

[14] P.Adamson et al., The NuMI neutrino beam, 2016 DOI: 10.1016/j.nima.2015.08.063

[15] Jianming Bian, The NOvA Experiment: Overview and Status, arXiv:1309.7898

[16] Luke Vinton, Calibration Technotes, `NOVA-doc-13579-v35`

[17] K.A. Olive et al. (Particle Data Group), Chin. Phys. C, 38, 090001 (2014).

[18] Adam Aurisano, The NOvA Detector Simulation, `NOVA-doc-13577-v3`

[19] C.Andreopoulos et al., The GENIE Neutrino Monte Carlo Generator, Nucl.Instrum.Meth.A614 (2010) 87-104

[20] S. Agostinelli et al., Geant4 - A Simulation Toolkit, Nucl. Instrum. Meth. A 506 (2003) 250-303

[21] C. Chou, The Nature of the Saturation Effect of Fluorescent Scintillators, Phys. Rev. 87, 904 (1952).

[22] Adam Aurisano, Tech Note: 2017 Light Model, `NOVA-doc-23228-v2`

[23] Martin Ester and Hans-Peter Kriegel and Jörg Sander and Xiaowei Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pages 226–231

[24] Michael Baird, Tech Note for the Multi-Hough Transform, NOVA-doc-8241-v1

[25] Mark Messier, Vertex reconstruction based on elastic arms, NOVA-doc-7530-v4

[26] M. Gyulassy and M. Harlander, Computer Physics Communications, 66, (1991) 32-46.

[27] Martin Pincus, A Monte Carlo Method for the Approximate Solution of Certain Types of Constrained Optimization Problems, 1970 DOI: 10.1287/opre.18.6.1225

[28] Evan Niner, Technical note of FuzzyKVertex reconstruction, NOVA-doc-7648-v2

[29] R. Krishnapuram, J.M. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Syst. 1 (1993) 98110.

[30] Dunn, J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, DOI: 10.1080/01969727308546046

[31] Nicholas J Raddatz, KalmanTrack Technical Note, NOVA-doc-13545-v1

[32] Hamilton, Time Series Analysis, Princeton University Press. Chapter 13, 'The Kalman Filter'

[33] Michael Baird, Break Point Fitter Technical Note, NOVA-doc-32455-v1

[34] Nicholas J Raddatz, ReMId Technical Note, NOVA-doc-11206-v1

[35] Altman, Naomi S., An introduction to kernel and nearest-neighbor nonparametric regression, DOI: 10.1080/00031305.1992.10475879

[36] Jianming Bian, Evan Niner, Kanika Sachdev, LID and e/pi0 Identifier technote, NOVA-doc-15344-v1

[37] Micah Groh, Ryan Murphy, Fernanda Psihas, Prong CVN Performance Comparison, NOVA-doc-29519-v2

[38] Szegedy, Christian et al., Going Deeper with Convolutions, arXiv:1409.4842

[39] Adam Aurisano et al., A Convolutional Neural Network Neutrino Event Classifier, NOVA-doc-14905-v6

[40] Kirk Bays, Cosmic rejection technical note, NOVA-doc-11205-v5

[41] Schapire, Robert; Singer, Yoram, Improved Boosting Algorithms Using Confidence-rated Predictions, CiteSeerX: 10.1.1.33.4002

[42] Susan Lein, Numu Energy Estimators - SA Technote, NOVA-doc-15211-v1

[43] Shiqi Yu, Fernanda Psihas, Zelimir Djurcic, Technote: Nue Energy Estimator 2017, NOVA-doc-22502-v14

[44] Erika Catano-Mur, Bannanje Nitish Nayak, Andrew Thomas Cleve Sutton, Thomas Warburton , Event selection for the 2020 3-flavor analysis, NOVA-doc-44040-v2

[45] Diana Patricia Mendez Mendez, Technote: Sensitivities for the FHC+RHC Disappearance Analysis, NOVA-doc-27404-v6

[46] Ashley Back, Reddy Pratap Gandrajula, Lyudmila Kolupaeva, Daniel Pershey, Nue selection and binning for the 2018 analysis, NOVA-doc-27290-v2

[47] Erika Catano-Mur, BEN decomposition technote 2017, NOVA-doc-23174-v2

[48] Lyudmila Kolupaeva, Extrapolation and event prediction in the 2018 nue analysis, NOVA-doc-28208-v1

[49] Erika Catano-Mur, Numu sensitivity technote (2017), NOVA-doc-23874-v2

[50] Erika Catano-Mur, Daniel Pershey, Technical Note for Fitting, Feldman-Cousins, and Sensitivities for the 2017 Joint Analysis, NOVA-doc-23126-v3

[51] Particle Data Group, Review of Particle Physics, DOI:10.1103/PhysRevD.98.030001

[52] McCulloch, Warren; Walter Pitts (1943). A Logical Calculus of Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics. 5 (4): 115–133., DOI:10.1007/BF02478259

[53] Christopher M. Bishop, Pattern Recognition and Machine Learning, ISBN-13: 978-0387-31073-2

[54] Schmidhuber J., Deep learning in neural networks: An overview, Neural networks, 61, 85-117, arXiv:1404.7828

[55] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E., ImageNet classification with deep convolutional neural networks, DOI:10.1145/3065386

[56] Sepp Hochreiter; Jurgen Schmidhuberk, Long short-term memory. DOI:10.1162/neco.1997.9.8.1735

[57] Alexander Radovic, LSTMs For Energy Estimation a Proof of Concept, NOVA-doc-18801-v1

[58] Dmitrii Torbunov, LSTM NuMu Energy Estimator, Technical Details , NOVA-doc-33719-v3

[59] Dmitrii Torbunov, Prod 3, FHC FD Energy Estimator, Periods 1,2 Discrepancy, v2, NOVA-doc-19850-v2

[60] Dmitrii Torbunov, LSTM NuMu Energy Estimator, Sensitivity to Cuts and Input Perturbations, NOVA-doc-35296-v1

[61] Dmitrii Torbunov, LSTM NuMu Energy Estimator, Progress Report, NOVA-doc-35766-v1

[62] Dmitrii Torbunov, LSTM NuMu Energy Estimator, Status Update, NOVA-doc-33884-v2

[63] Dmitrii Torbunov, LSTM NuMu Energy Estimator, ND Weights Tune, NOVA-doc-40351-v1

[64] Dmitrii Torbunov, LSTM NuMu Energy Estimator, Status Update, NOVA-doc-34564-v1

[65] Dmitrii Torbunov, LSTM NuMu Energy Estimator, Noise vs Systematics, NOVA-doc-40032-v1

[66] Dmitrii Torbunov, LSTM NuMu Energy Estimator, Noise Models, Sensitivity Contours, NOVA-doc-39837-v1

[67] Dmitrii Torbunov, Required Systematics for the LSTM EE for Neutrino 2022, NOVA-doc-47796-v1

[68] Quanshi Zhang, Ying Nian Wu, Song-Chun Zhu, Interpretable Convolutional Neural Networks, arXiv:1710.00935

[69] Gavin S Davies et al., CVN 2018 Technical note, NOVA-doc-27467-v4

[70] Andrew Vold, Improving Physics Based Electron Neutrino Appearance Identification with a Long Short-Term Memory Network, NOVA-doc-32425-v1

[71] Dmitrii Torbunov, Slice LID, Initial Studies, NOVA-doc-35769-v1

[72] Dmitrii Torbunov, Slice LID, Model Complexity Studies, NOVA-doc-36690-v1

[73] Dmitrii Torbunov, SliceLID, Status and Plans, NOVA-doc-37185-v1

[74] Dmitrii Torbunov, Slice LID, Art Integration Update, NOVA-doc-37689-v1

[75] Laurens van der Maaten and Geoffrey Hinton, Visualizing Data using t-SNE, 2008, Journal of Machine Learning Research. 9: 2579–2605

[76] NOvA Collaboration, First Measurement of Neutrino Oscillation Parameters using Neutrinos and Antineutrinos by NOvA, DOI: 10.1103/PhysRevLett.123.151803

[77] Ryan B. Patterson, Matthew J. Wetstein, Internal NOvA Document, NOVA-doc-36936-v2

[78] Dmitrii Torbunov, LSTM NuMu Energy Estimator with Systematics (short version), NOVA-doc-47453-v1

[79] Dmitrii Torbunov, LSTM NuMu Energy Estimator with Systematics, NOVA-doc-47438-v1

[80] Shaowei Wu, Systematics of LSTM energy estimator, NOVA-doc-48110-v1

[81] Dmitrii Torbunov, LSTM Energy Estimator Documentation, NOVA-doc-45821-v3

[82] Dmitrii Torbunov, SliceLID, Performance Studies, NOVA-doc-47051-v1

[83] Dmitrii Torbunov, SliceLID Documentation, NOVA-doc-46199-v1

Appendix A

LSTM EE Results on Real Data

In section 7.2 and section 8.1 I have shown the NuMu sensitivity contours made with the simulated Far Detector events using the LSTM energy estimator. Such contours are sufficient to understand the performance improvements that the LSTM EE could provide. However, it is customary in the Ph.D. dissertations to make the NuMu contours on the real Far Detector data (called data contours). In this appendix, I will show the NuMu data contours made on the real Far Detector data, using the LSTM EE in place of the standard NuMu energy estimator.

As was discussed during the development of the LSTM EE, there are two systematic uncertainties that need to be modified to work with the LSTM EE. Since NOvA has not found a proper way to perform such modifications yet I will exclude these systematics (Muon Range and Neutron systematics) from the analysis chain.

A.1 Data/MC Comparison at the Near Detector

I will begin this appendix by comparing the LSTM energy distributions at the Near Detector between Data and MC. These energy distributions should match within uncertainties. If the distributions do not match, then it will indicate issues with the NOvA simulation. In particular, it might mean that the LSTM EE couples strongly to some feature that is present in the simulation but absent in the real data, and therefore the LSTM EE would need to be modified not to use such a feature.

I have run the official NOvA scripts to create the Data/MC comparison plots

at the Near Detector and the result is shown in Figure A.1. Figure A.1 shows that there is a rather good agreement between the reconstructed energy distributions in Data and MC.

Since Figure A.1 shows good agreement between Data and MC, it means that there is no immediate problem with the LSTM EE coupling to the unphysical features present in the simulation.

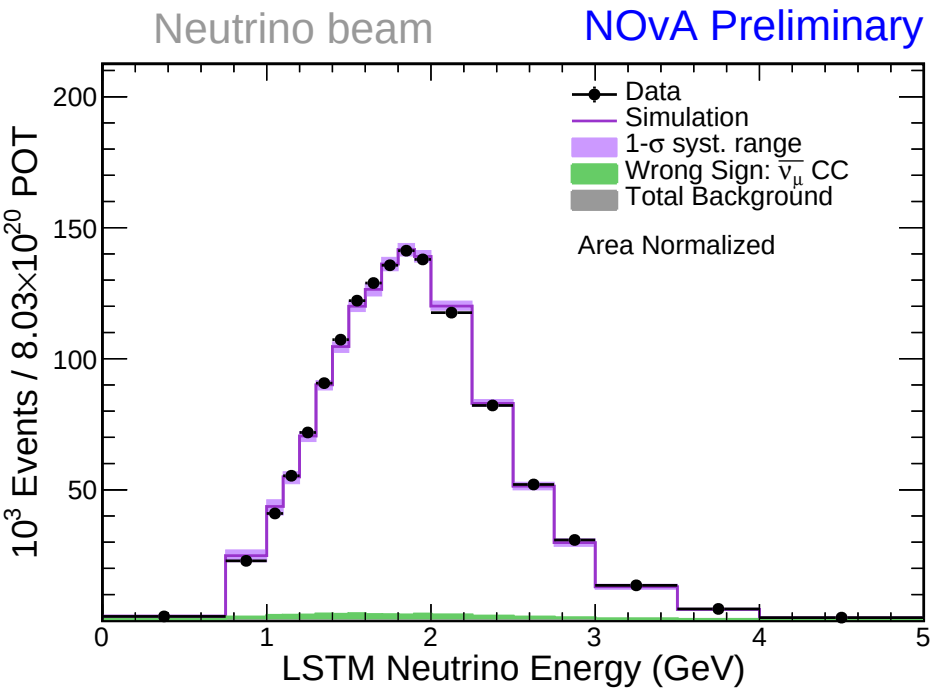


Figure A.1: Data/MC comparisons of the reconstructed energy distributions predicted by the LSTM EE at the Near Detector. The black dots represent the real data points, their error bars show the statistical uncertainty. The violet histogram is made on MC, its error margin represents the systematic uncertainty.

A.2 Data Contours

After verifying that the LSTM energy estimator does not show obvious issues at the Near Detector, I have run the full analysis chain to obtain the NuMu data contours. Unfortunately, the NOvA optimizer failed to fit the NuMu data surface for the LSTM

EE with the Light-Level systematic. Therefore, in addition to excluding the Muon Range and Neutron systematics, I have excluded the Light-Level systematic from the analysis as well.

Figure A.2 shows the NuMu data contours obtained with the LSTM energy estimator. To assess the degree of improvement that the new energy estimator brings, we can compare this plot to the NuMu data contours made with the standard spline-based energy estimator shown in Figure A.3. The LSTM EE contours are narrower than the Spline EE contours, which means that the LSTM EE gives better precision of measurements of the neutrino oscillation parameters.

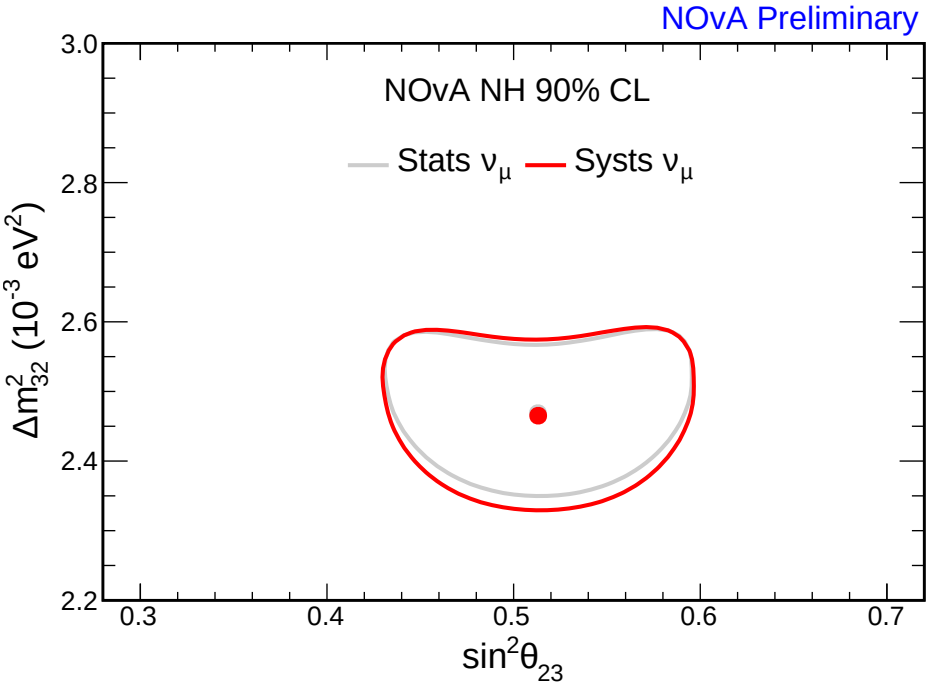


Figure A.2: NuMu data contours made with the LSTM EE on the FHC dataset.

We can also obtain estimates of the neutrino oscillation parameters and their confidence intervals in the Gaussian limit (c.f. section 6.5). Usage of the LSTM EE gives estimates of the neutrino oscillation parameters summarized in Equation A.1. The estimates of the neutrino oscillation parameters obtained with the Spline EE

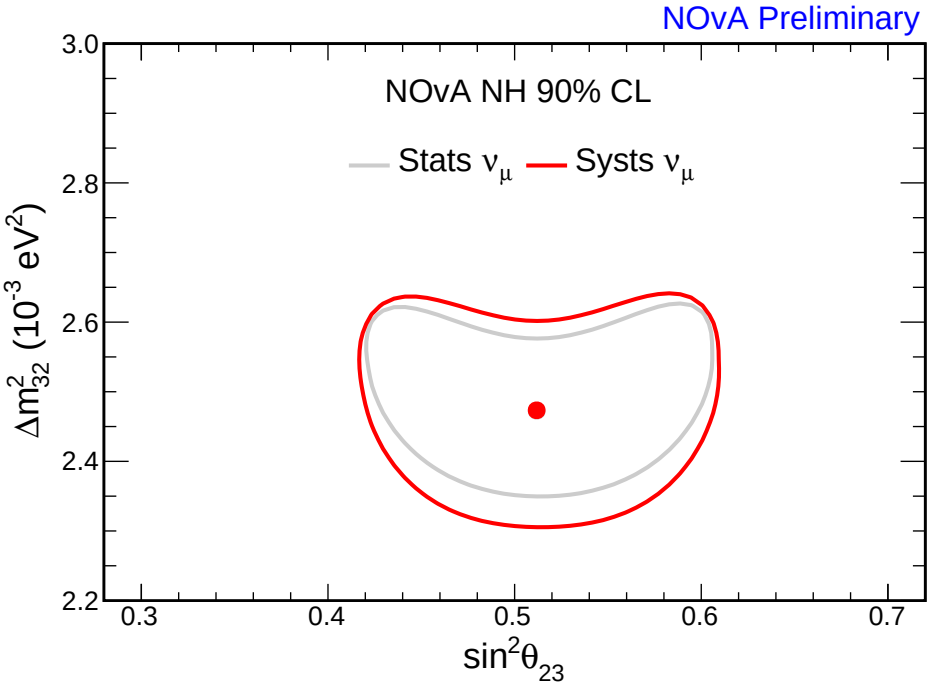


Figure A.3: NuMu data contours made with the Spline EE on the FHC dataset.

are shown in Equation A.2.

$$\Delta m^2_{32} = 2.47^{+0.05}_{-0.07} \times 10^{-3}\ \text{eV}^2,\ \sin^2\theta_{23} = 0.51^{+0.05}_{-0.05} \tag{A.1}$$

$$\Delta m^2_{32} = 2.47^{+0.07}_{-0.09} \times 10^{-3}\ \text{eV}^2,\ \sin^2\theta_{23} = 0.51^{+0.06}_{-0.06} \tag{A.2}$$

Comparing the estimates of the neutrino oscillation parameters between the LSTM EE (A.1) and the Spline EE (A.2) we see that the LSTM EE estimates are more precise, especially for the Δm^2_{32} parameter.

Finally, Figure A.4 shows the predicted and observed Far Detector energy spectrum obtained with the LSTM EE. Figure A.5 shows the same spectrum, but separated per the hadronic energy quartiles (c.f. subsection 6.2.1).

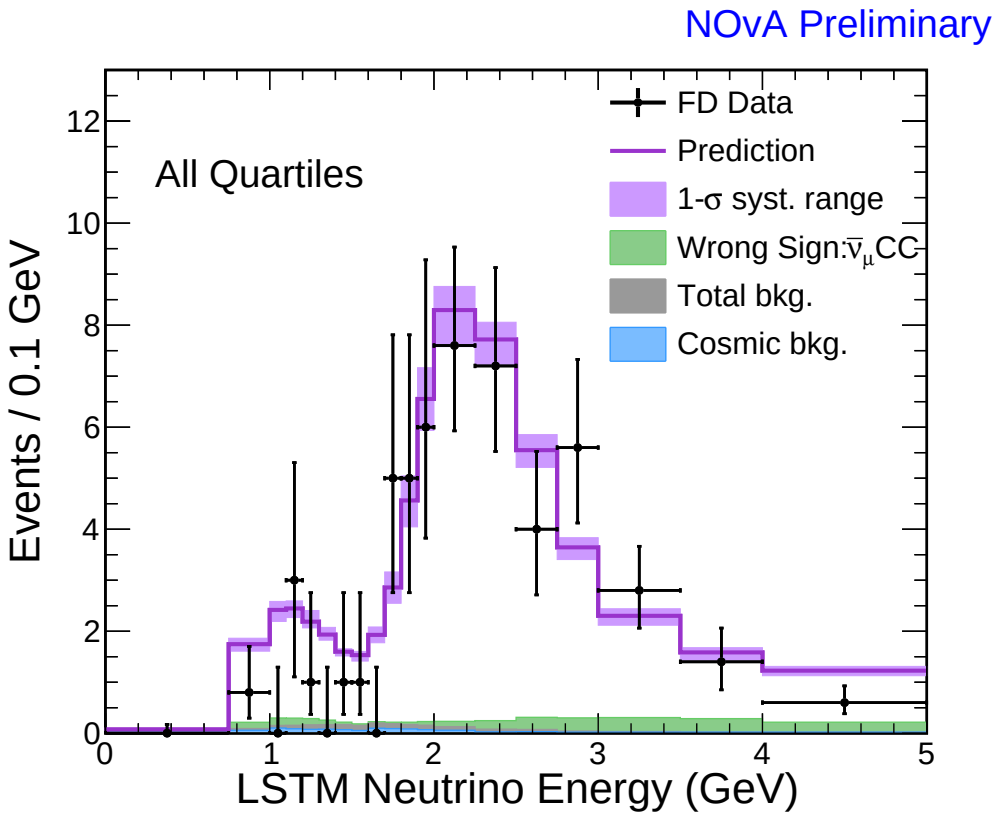


Figure A.4: Comparison of the predicted Far Detector energy spectrum and the observed one for the LSTM EE.

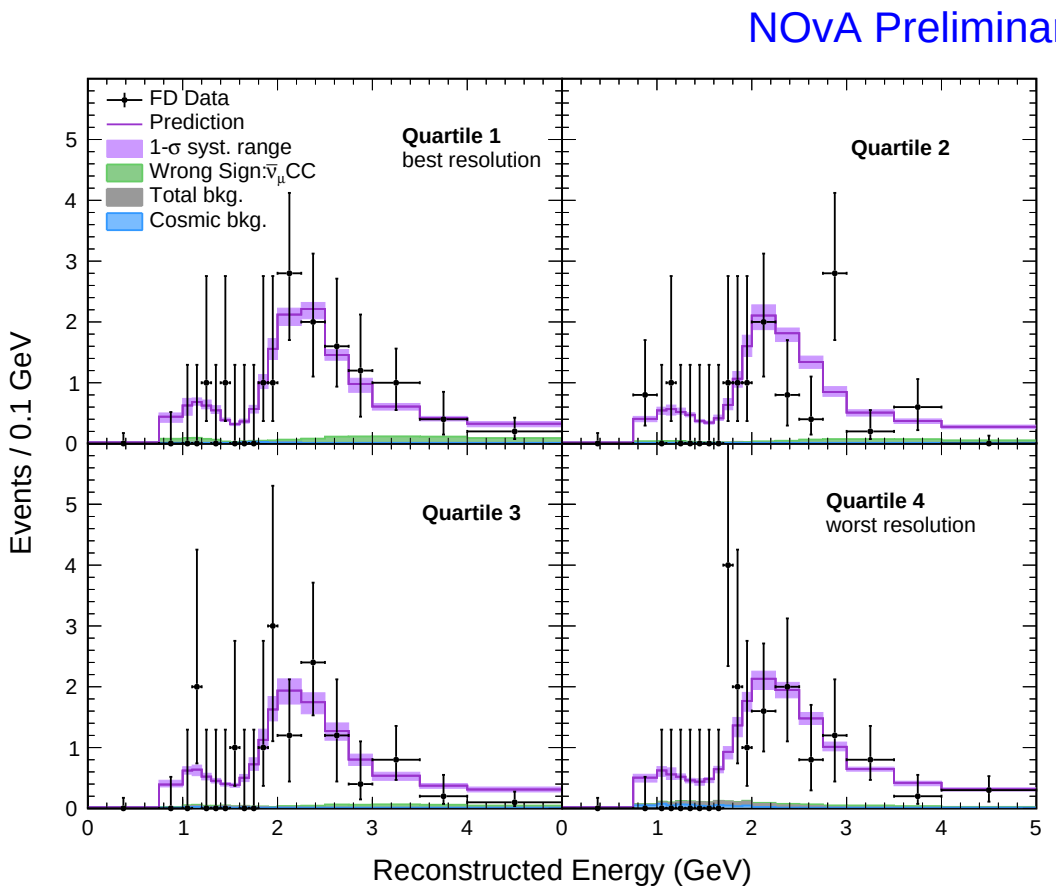


Figure A.5: Comparison of the predicted Far Detector energy spectra and the observed ones for the LSTM EE separated by the hadronic energy quartiles.